

8-2013

# APPLICATION OF PARAMETER ESTIMATION AND CALIBRATION METHOD FOR CAR-FOLLOWING MODELS

Md Rahman

Clemson University, [mdr@clemson.edu](mailto:mdr@clemson.edu)

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_theses](https://tigerprints.clemson.edu/all_theses)



Part of the [Civil Engineering Commons](#)

---

## Recommended Citation

Rahman, Md, "APPLICATION OF PARAMETER ESTIMATION AND CALIBRATION METHOD FOR CAR-FOLLOWING MODELS" (2013). *All Theses*. 1763.

[https://tigerprints.clemson.edu/all\\_theses/1763](https://tigerprints.clemson.edu/all_theses/1763)

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

APPLICATION OF PARAMETER ESTIMATION  
AND CALIBRATION METHOD FOR  
CAR-FOLLOWING MODELS

---

A Thesis  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
Civil Engineering

---

by  
Md. Mizanur Rahman  
August 2013

---

Accepted by:  
Dr. Mashrur A. Chowdhury, Committee Chair  
Dr. Taufiqaur Khan  
Dr. Jennifer Ogle

## ABSTRACT

Both safety and the capacity of the roadway system are highly dependent on the car-following characteristics of drivers. Car-following theory describes the driver behavior of vehicles following other vehicles in a traffic stream. In the last few decades, many car-following models have been developed; however, studies are still needed to improve their accuracy and reliability.

Car-following models are a vital component of traffic simulation tools that attempt to mimic driver behavior in the real world. Microscopic traffic simulators, particularly car-following models, have been extensively used in current traffic engineering studies and safety research. These models are a vital component of traffic simulation tools that attempt to mimic real-world driver behaviors. The accuracy and reliability of microscopic traffic simulation models are greatly dependent on the calibration of car-following models, which requires a large amount of real world vehicle trajectory data.

In this study, the author developed a process to apply a stochastic calibration method with appropriate regularization to estimate the distribution of parameters for car-following models. The calibration method is based on the Markov Chain Monte Carlo (MCMC) simulation using the Bayesian estimation theory that has been recently investigated for use in inverse problems. This dissertation research includes a case study, which is based on the Linear (Helly) model with a different number of vehicle trajectories in a highway network.

The stochastic approach facilitated the calibration of car-following models more realistically than the deterministic method, as the deterministic algorithm can easily get stuck at a local minimum. This study also demonstrates that the calibrated model yields smaller errors with large sample sizes. Furthermore, the results from the Linear model validation effort suggest that the performance of the calibration method is dependent upon size of the vehicle trajectory.

## DEDICATION

*I dedicate this thesis to my parents in recognition of their love and inspiration.*

## ACKNOWLEDGEMENTS

I wish to take this opportunity to thank all the people responsible for my successful graduation from the Glenn Department of Civil Engineering at Clemson University.

First, I would like to thank my advisor, Professor Dr. Mashrur Chowdhury, for his faith in me. His sheer brilliance, sound technical insights and constant support were highly inspiring. I feel privileged to know him and to be a part of his research group. I also would like to thank my committee member Dr. Taufiqur Khan his guidance and valuable support and Dr. Jennifer Ogle for taking the time to review my thesis.

I greatly enjoyed the coursework within and beyond the department. It contributed greatly to an educationally enriching experience. Without doubt, my lab partners exerted the greatest influence on my life at Clemson University. I was lucky to be a part of very congenial lab environment, something I consider fundamental for a stress-free work environment. I am indebted to them for their support and friendship. I especially thank Yiming and Thilo for supporting me in the technical aspects of my research.

I would like to thank Akhter and Kakan for providing focus and inspiration at the early stages of my academic career at Clemson University. . My friends at Clemson University (too many to mention by name) made this place a home away from home. I never ever felt out of my comfort zone at any point of my stay at Clemson University, and the credit goes to a highly jovial and friendly Bangladeshi Community that I was lucky to be a part of. My roommates, Arif and Jaman, provided great support during my tough times, and I am very grateful for their kindness. I also thank the staff of the Civil

Department, and in particular Kristi and Monica, for clearing away all the administrative issues with an ever-present smile.

Above all, I thank my family, who define me and responsible for whatever I have achieved so far in my life.

# TABLE OF CONTENTS

	Page
<b>TITLE PAGE.....</b>	<b>I</b>
<b>ABSTRACT.....</b>	<b>II</b>
<b>DEDICATION.....</b>	<b>IV</b>
<b>ACKNOWLEDGEMENTS .....</b>	<b>V</b>
<b>TABLE OF CONTENTS .....</b>	<b>VII</b>
<b>LIST OF TABLES .....</b>	<b>IX</b>
<b>LIST OF FIGURES .....</b>	<b>X</b>
<b>NOTATIONS.....</b>	<b>XII</b>
 <b>CHAPTER ONE .....</b>	 <b>1</b>
INTRODUCTION	
1.1 Overview of Calibration of Car-Following Models.....	1
1.2 Parameter Estimation of Car-Following Models .....	2
1.3 Motivation for New Calibration Method .....	3
1.4 Objectives of the Research.....	4
1.5 Organization of Thesis.....	4
 <b>CHAPTER TWO .....</b>	 <b>5</b>
REVIEW OF CAR-FOLLOWING AND LANE-CHANGING MODELS	
2.1 Introduction.....	5
2.2 Car-Following Models .....	5
2.3 Lane-Changing Models.....	11
2.4 Summary of Literature Review .....	43
 <b>CHAPTER THREE.....</b>	 <b>44</b>
CALIBRATION METHOD	
3.1 Introduction.....	44
3.2 General Calibration Method .....	44
3.3 General Framework for Validation of the Calibration Method .....	51



<b>CHAPTER FOUR.....</b>	<b>53</b>
MODEL CALIBRATION AND VALIDATION RESULTS	
4.1 Introduction.....	53
4.2 Dataset description.....	53
4.3 Model Parameter Estimation.....	55
4.4 Calibration Results.....	70
4.5 Evaluation .....	71
4.6 Validation Results .....	74
4.7 Contribution of the Research .....	77
 <b>CHAPTER FIVE .....</b>	 <b>78</b>
CONCLUSIONS AND RECOMMENDATIONS	
5.1 Conclusions.....	78
5.2 Recommendations .....	80
 <b>REFERENCES.....</b>	 <b>81</b>
 <b>APPENDICES.....</b>	 <b>88</b>
Appendix A.....	89
Appendix B .....	95
Appendix C .....	103

## LIST OF TABLES

Table		Page
2.1	Theoretical Comparison of Lane-Changing Model Categories.....	41-41
4.1	Validation of Calibration Method.....	63
4.2	Model Parameters Summary.....	71
4.3	Average Mean Square Error (MSE) per Vehicle for Calibration .....	72
4.4	Average Mean Square Error (MSE) per Vehicle for Validation.....	74

## LIST OF FIGURES

Figure		Page
2.1	Classification of Lane-Changing Models.....	13
2.2	Pay-Off Matrices for each Player.....	24
3.1	Calibration and Validation Steps of a Car-Following Model .....	44
3.2	Bayesian Framework for Calibration of Car-Following Model.....	45
3.3	Definition of Inverse Problem.....	46
3.4	Bayes Estimation Process using MCMC Method (Metropolis-Hastings Algorithm).....	50
3.5	Validation of Bayesian Framework Calibration Method of Car-Following Model.....	52
4.1	Study Area Schematic.....	54
4.2	Histogram of Parameter for Five Hundred Vehicles using Assumed Distribution of Parameter.....	58
4.3	Convergence of parameter with synthetic data of five hundred vehicle trajectories using normal prior distribution.....	59
4.4	Histogram of parameter with synthetic data of five hundred vehicle trajectories using normal prior distribution.....	60
4.5	Histogram of parameter with synthetic data of five hundred vehicle trajectories using uniform prior distribution.....	61

## List of Figures (Continued)

Figure		Page
4.6	Convergence of parameter with ten (10) vehicle trajectories.....	66
4.7	Histogram of parameter with ten (10) vehicle trajectories.....	67
4.8	Histogram of parameter with five hundred (500) vehicle trajectories.....	68
4.9	Histogram of parameter with one thousand (1000) vehicle trajectories.....	69
4.10	Comparison of acceleration/deceleration profile among estimated data and observed data for calibration.....	73
4.11	Comparison of acceleration/deceleration among predicted data and observed data for validation.....	76

## NOTATIONS

$C$	Velocity transformer to an observable quantity
$v(q)$	Observed field data
$q$	The vector of model parameters of length $k$
$k$	Number of element of vector parameter
$P(q)$	The prior distribution of $q$
$d$	Actual distribution of $m$
$P(d/q)$	Conditional probability of the observation given the cause $q$
$P(q/d)$	Conditional probability of possible cause given that some effect has been observed
$n$	Follow vehicle
$n-1$	Lead vehicle
$a_n(t)$	Acceleration of vehicle $n$ at time $t$
$a_{n-1}(t-T)$	Acceleration of vehicle $n-1$ at time $t-T$
$D_n(t)$	Desired distance factor
$v_n(t)$	Speed of vehicle $n$ at time $t$
$v_{n-1}(t)$	Speed of vehicle $n-1$ at time $t$
$v_{n-1}(t-T)$	Speed of vehicle $n-1$ at time $t-T$
$\Delta v(t-T)$	Speed difference respectively of vehicle $n$ and $n-1$
$\Delta x(t-T)$	Position difference respectively of vehicle $n$ and $n-1$

# CHAPTER ONE

## INTRODUCTION

### 1.1 Overview of Calibration of Car-Following Models

As one of the fundamental concepts in transportation, the car-following theory includes the essential element of a traffic stream, the behavior of one vehicle following a preceding vehicle. Such behavior is a cumulative outcome of a series of factors, such as the psychological and physical state of the driver, the conditions of the traffic stream, and the performance of the vehicle. Therefore, the car-following theory has received attention in areas such as human factors, traffic flow theory and vehicle dynamics. With the higher computing capacity available today, a number of widely used traffic simulation tools utilize various car-following theories to mimic microscopic interactions between vehicles (Brackstone and McDonald, 1999). The automotive industry, which first initiated the development of car-following models for vehicle design purposes (Chandler et al., 1958), also studied and utilized car-following theory for a variety of other reasons, such as for understanding the human factor in Adaptive Cruise Control (Vahidi and Eskandarian, 2003). Work on car-following theory can be traced back to 1950s (Brackstone and McDonald, 1999).

Recently, more attention has been given to the calibration of car-following models, especially with real-world data. By utilizing modern sensing, tracking and data collection technologies, vehicle trajectory data, which is important for the calibration of car-following models, can be obtained with high accuracy. For instance, under the Next Generation Simulation (NGSIM) project, high-resolution vehicle trajectory data was

collected with a digital video camera and analyzed using vehicle detection/tracking software for several freeways, highways and urban arterials (Interstate 80 Freeway Dataset, 2006).

Several studies attempted to calibrate car-following models with different data sets. Schultz and Rilett (Schultz and Rilett, 2004) examined a methodology to obtain the calibration factors of car-following models from the distribution of a parameter obtained through microscopic traffic simulation. The application of such methodology on IH-10 eastbound in Houston, Texas indicated the efficacy of the methodology on the macroscopic level; however the performance in the microscopic level is unclear. Kesting and Treiber (2008) calibrated two car-following models with empirical vehicle trajectories by minimizing the error between trajectory data and the values predicted by the models. This study yielded errors that ranged from 11% to 29% and also indicated that intra-driver variability accounts for a larger part of the error than inter-driver variability does. Hoogendoorn and Hoogendoorn (2010) proposed a genetic calibration framework that can estimate the parameters statistically by utilizing multiple trajectories simultaneously.

## **1.2 Parameter Estimation of Car-Following Models**

Recently, few researchers studied parameter estimation of car-following models, in which parameter estimation of various car-following models was conducted with a deterministic framework. The model parameters were estimated by minimizing a cost function, i.e., the output least square with a data fitting term and a Tikhonov regularization term. This method guarantees a solution near the nominal value of the parameters. Minimization techniques, particularly the Levenberg-Marquardt method,

were used to solve for the minimum norm solution near a nominal value. A deterministic approach can estimate an average parameter value; however, often the optimization routine might get stuck at a local minimum. Global deterministic minimization methods for parameter estimation of car-following models are very challenging due to the ill-posed nature of the inverse problem. In deterministic methods, the minimization approach involved estimating the parameters primarily to find the average value of the parameters.

### **1.3 Motivation for New Calibration Method**

The calibration process using the deterministic minimizing approach involved estimating the parameters to find the average value of the parameters. On the other hand, if one can estimate the distribution of each of the car-following model parameters, then the aggregate behavior of a large number of cars can be better simulated. The statistical parameter estimation approach can also quantify uncertainty in the parameters, which can be particularly useful for calibration purposes.

Therefore, in this thesis, a Bayesian framework is developed with appropriate regularization for estimation of the statistical distribution of the parameters of car-following models. Then in order to prove the efficacy of the proposed approach, a Bayesian framework was applied to a specific car-following model to provide a comparison with the deterministic approach. Current work presented in this thesis is based on the Markov chain Monte Carlo (MCMC) method that uses Bayesian estimation theory. Bayesian estimation theory has been recently investigated for inverse problems (Kaipio and Somersalo, 2005).



## **1.4 Objectives of the Research**

The objective of this research is to develop a process for applying a stochastic calibration method to estimate parameters of linear car-following models and to apply the process to a car-following model as a case study. In order to prove the efficacy of the proposed approach, a synthetic dataset was used to estimate the parameters of a linear model with both normal and uniform prior distribution of the parameters. The calibration method was then applied to a relatively simple car-following model: the linear (Helly) model. In addition, this thesis includes a validation of the calibrated car-following model with real world vehicle trajectory data.

## **1.5 Organization of Thesis**

Chapter 2 presents a review of different car-following and lane-changing models and summarizes the major parameters considered in each of these models, while Chapter 3 outlines the method for the development of the process of the stochastic calibration and validation. In Chapter 4, the results of detailed statistical analysis are presented. Finally, Chapter 5 presents conclusions and recommendations.

## CHAPTER TWO

### REVIEW OF CAR-FOLLOWING AND LANE-CHANGING MODELS

#### 2.1 Introduction

Commonly known car-following models can be classified into five common groups: the Gazis-Herman-Rothery (GHR) model, the Collision Avoidance (CA) model, the Linear Model, the Fuzzy-logic-based model, and the Optimal Velocity (OV) model and its variations (Panwai and Dia, 2005 and Brackstone and McDonald, 1999). Car-following models along with their advantages and disadvantages are discussed in this chapter.

Although lane-changing models are out of the analysis scope of this thesis, car-following and lane-changing models collaboratively describe traffic flow at both microscope and macroscopic levels. As lane-changing models are an integral part of traffic flow along with car-following models, this chapter also discusses existing lane-changing models.

#### 2.2 Car-Following Models

Car-following models mathematically describe the behavior by which drivers follow the preceding vehicle in a traffic stream (Brackstone and McDonald, 1999). These mathematical expressions are validated and refined with collected traffic measurements.

Within the traffic stream, a driver's reaction time is defined as the elapsed time between any changes made in the predecessor vehicle and the driver's subsequent

response in reacting to changing headway. In other words, this reaction time is caused by the fact that a certain amount of time elapses before the driver notices the difference. By integrating the car-following behavior with respect to time, the behavior of the traffic stream can be presented by individual driver responses. This integration also serves as the foundation of traffic flow diagrams, the flow rate  $q$  and the speed  $u$  as a function of the density  $k$  (Reijmers, 2006). Therefore, the formulas used to describe the behavior of individual drivers can be used to derive a criterion for the stability of the traffic stream. The basic formula to describe the reaction time of the individual car-following behavior is as follows (Reijmers, 2006):

$$Reaction(t+T) = Sensitivity \times Stimulus(t)$$

Based on the sensitivity of each individual driver, a reaction to a stimulus at time  $t$  occurs after a reaction time  $T$ , resulting in acceleration or deceleration of the vehicle. The driving task is relatively easy when there is no preceding vehicle; the driver just needs to maintain his or her desired speed. When a preceding vehicle exists, however, the driver needs to keep a desired distance (headway) between vehicles, which is related to the speed of the vehicle and the speed difference between vehicles. The driver controls the vehicle via accelerating or braking, changing speed with respect to time ( $\frac{dv}{dt}$ ) in other words, with his/her perception. Therefore, the car-following model can be expressed as

$$\frac{dv}{dt} = f(v, \Delta x, \Delta v).$$

There are six commonly used car-following models, including: Gazis-Herman-Rothery (GHR) model, Collision Avoidance (CA) model, Linear Model, Fuzzy-logic-

based model, Optimal Velocity (OV) model and Meta models. The following section summarizes the major parameters considered in each of these models.

### 2.2.1 Gazis-Herman-Rothery (GHR) model

The GHR model is one of the oldest and most well-studied models, the basic formula of which is shown in below.

$$a_n(t) = cv_n^m(t) \frac{\Delta v(t - T)}{\Delta x^l(t - T)}$$

In the formula,  $a_n(t)$  and  $v_n(t)$  are the acceleration and velocity for vehicle  $n$  at time  $t$  respectively;  $\Delta v$  and  $\Delta x$  are the speed and position difference respectively of vehicle  $n$  and  $(n-1)$ ;  $T$  is the reaction time of driver;  $c$ ,  $m$  and  $l$  are the parameters to be calibrated.

Many works have been undertaken on calibrations (Brackstone and McDonald, 1999) since 1958. After 1972, most of these calibrations were performed for specific traffic or driving conditions because these parameters were known to likely vary between different conditions. This variance is also one of the reasons for the lack of further investigation on the GHR model, especially after 2000.

### 2.2.2 Collision Avoidance (CA) model

The mathematical formulation of Collision Avoidance (CA) model proposed by Kometani and Sasaki (Kometani and Sasaki, 1959) is shown in below.

$$\Delta x(t - T) = \alpha v_{n-1}^2(t - T) + \beta_1 v_n^2(t) + \beta v_n(t) + b_0$$

Where  $v_n$  and  $v_{n-1}$  are the speed of vehicle  $n$  and  $(n-1)$ ;  $\Delta x$  is the relative distance between vehicle  $n$  and  $(n-1)$ ;  $T$  is the reaction time;  $\alpha$ ,  $\beta_1$ ,  $\beta$  and  $b$  are the constants to be calibrated.

Of the several variations reported (Panwai and Dia, 2005), the Gipps model (Gipps, 1981) is perhaps the most important and widely used simulation model (Panwai and Dia, 2005 and Brackstone and McDonald, 1999).

### 2.2.3 Linear (Helly) Model

The Linear, or Helly model, which was developed from the GHR model, is the third model shown in below.

$$a_n(t) = C_1 \Delta v(t - T) + C_2 \{\Delta x(t - T) - D_n(t)\}$$

$$D_n(t) = \alpha + \beta v(t - T) + \gamma a_n(t - T)$$

Here,  $a_n(t)$  is the acceleration of vehicle  $n$ ;  $\Delta v$  and  $\Delta x$  are the speed and position difference respectively of vehicle  $n$  and  $(n-1)$ ;  $T$  is the reaction time;  $D_n(t)$  is the desired following distance. In addition,  $C_1$ ,  $C_2$ ,  $\alpha$ ,  $\beta$  and  $\gamma$  are the constants, whose calibration is the main difficulty of this model ((Panwai and Dia, 2005).

### 2.2.4 Fuzzy-Logic-Based Model

The application of the fuzzy logic model to the car-following theory occurred in the 1990s. The first attempt was to apply fuzzy rules on the GHR model Kikuchi and Chakroborty, 1992). This kind of model is unique because the human driver is a fuzzy system rather than a precise machine, and thus, more likely to represent real human driving behavior. However, it is difficult to calibrate the membership function, which is the most important part of the model (Brackstone and McDonald, 1999). Research has been conducted in this area. Brackstone et al. (Brackstone and McDonald, 2002) investigated this subject using the road subjectivity test and Chakroborty and Kikuchi

(Chakroborty and Kikuchi, 1999) calibrated the membership function in the fuzzy inference system by transforming it into an artificial neural network.

### 2.2.5 Optimal Velocity (OV) Model and Its Variations

Although the OV model, created by Bando et al. (Bando et al., 1995) in 1995, has been in existence for almost two decades, the real promise of this model has only recently been realized. In the original OV model, the acceleration of a vehicle is the function of the difference from the optimal speed and driver sensitivity.

$$\frac{dv_{n(t)}}{dt} = k[V(\Delta x_n(t)) - v_n(t)]$$

In the formula,  $v_n$  is the speed of vehicle  $n$ ;  $k$  is the sensitivity of the driver; and  $V$  is the OV function suggested by Helbing and Tilch (Helbing and Tilch, 1998). This is expressed as

$$V(\Delta x) = V_1 + V_2 \tanh[C_1 (\Delta x - l_c) - C_2]$$

where  $\Delta x$  is the relative distance between vehicle  $n+1$  and  $n$ ;  $l_c = 5\text{m}$  is the length of the vehicle;  $V_1 = 6.75 \text{ m/s}$ ,  $V_2 = 7.91 \text{ m/s}$ ,  $C_1 = 0.13 \text{ m}^{-1}$  and  $C_2 = 1.57$ . The OV model is unique in that it represents real traffic flow characteristics including stop-and-go traffic and the evolution of traffic congestion (Gong et al., 2008). However, unrealistic acceleration and deceleration also occurs when compared with field data (Peng and Sun, 2010). Basically, this model uses mathematical trigger functions because  $V$  is a phase transition model.

#### 2.2.5.1 Generalized Force (GF) Model

The GF model was developed by Helbing and Tilch (Helbing and Tilch, 1998) as the successor to the OV model. In this model, they use the negative speed difference, as shown below.

$$\frac{dv_{n(t)}}{dt} = k[V(\Delta x_n(t)) - v_n(t)] + \lambda H(-\Delta v_n(t))\Delta v_n(t)$$

Here,  $H$  is the Heaviside function;  $\lambda$  is a sensitivity coefficient;  $\Delta v$  is the speed difference of vehicle  $n$  and  $(n+1)$ . The change in the GF model improves data agreement.

#### 2.2.5.2 Full Velocity Difference (FVD) Model

The FVD model considers the full range of velocity difference rather than only the negative part. The model is shown below.

$$\frac{dv_{n(t)}}{dt} = k[V(\Delta x_{n(t)}) - v_{n(t)}] + \lambda \Delta v_n$$

This FVD model is an improvement over the GF model in that it contains a better description of the startup process (Jiang et al., 2001)

#### 2.2.6 Meta-models

Wiedemann (Wiedemann, 1974) created four categories or situations of driving (Treiber and Kesting, 2006) : Free Driving, Regulating, Stable Following and Braking. It is also called the Action Point (AP) model, assumes that a certain reaction will occur if a threshold is reached.

$$\frac{dv}{dt} = f(v, \Delta x, \Delta v)$$

In 2006, Treiber et al. (Treiber and Kesting, 2006) proposed a generalized model for all continuous models in which acceleration is a function of its own speed, relative distance and relative speed. This model consists of four elements: (1) finite reaction times, reaction time is not a multiple of the update time interval but a linear interpolation they proposed; (2) estimation errors, relative distance and relative speed are modeled as stochastic due to the differences in observation ability; (3) temporal anticipation, by being aware of their finite reaction time, drivers anticipate traffic conditions including future distance and future speed; (4) spatial anticipation, where the interactions between several vehicles downstream are considered.

## **2.3 Lane-Changing Models**

Driving tasks are conducted depending upon two fundamental considerations: maintaining a desired speed, and remaining in a lane for either downstream turning or passing maneuvers; the latter is usually described by lane-changing models mathematically. Lane-changing maneuvers consist of three critical driving behaviors: 1) lower-level control such as steering, acceleration, 2) monitoring which indicates awareness to maintain a situation, and 3) the decision to change lanes. The following sections summarize a detailed review of existing lane-changing models.

### 2.3.1 Classification of Lane-Changing Models

With the technological advancements for reliable traffic data collection, the lane-changing modeling has received increasing attention since the early 1980s ( Brackstone et al., 1998). The applications of lane-changing models can be broadly classified into two groups: adaptive cruise control and computer simulation. Lane-changing models for



adaptive cruise control are mainly focused on developing driving assistance models, which can be further classified into collision avoidance models and automation models. Collision avoidance models are for controlling drivers' lane-changing maneuvers and assisting them with completing lane changes safely. Automation models are for adjusting the steering wheel angle of vehicles automatically to perform safe lane-changing maneuvers (Lygeros et al. 1998; Nagel et al. 1998; Maerivoet and Moor 2005; Eidehall et al. 2007; Salvucci and Mandalia 2007; Doshi and Trivedi 2008; Kiefer and Hankey 2008; Li-sheng et al. 2009). Since the 1980s, many lane-changing models have been developed for micro-simulators to replicate driver decisions at the microscopic level. These lane-changing models are categorized into four groups: rule-based model, discrete choice-based model, artificial intelligence model, and incentive-based model (Figure 1). In the next four sections, the four types of microscopic lane-changing models are discussed in detail. Theoretical comparisons of these lane-changing models are presented in the following sections.

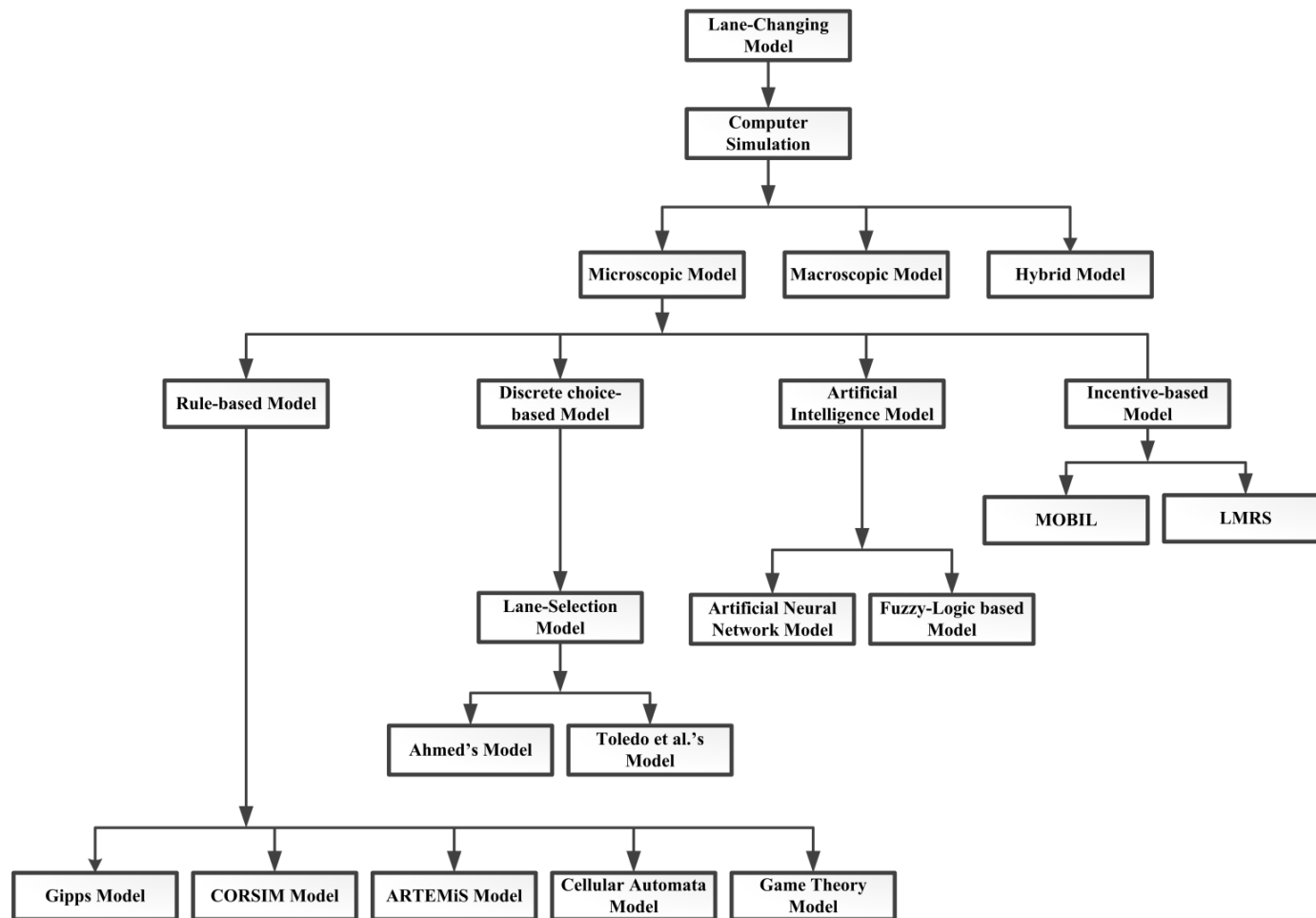


Figure 2.1: Classifications of Lane-Changing Models

### 2.3.2 Rule-based Models

#### 2.3.2.1 Gipps Model

Gipps model describes the lane-changing decisions and the execution of lane changes on freeways and urban streets as the result of three factors: lane-changing possibility, necessity for changing lanes and lane-changing desirability (Gipps, 1986). It incorporates the difference between the wish to change lanes and the execution of lane changes that was first introduced by Sparmann (Sparmann, 1978). Gipps model includes several factors, such as the existence of safety gap, locations of permanent obstructions, intent of turning movement, presence of heavy vehicles, and speed advantage. It also considers several lane-changing reasons: avoiding permanent obstructions, avoiding special-purpose lanes such as transit lanes, turning at downstream intersection, avoiding a heavy vehicle's influence, and gaining speed advantage. In this model, a driver's behavior falls into three zones, separated by the distance of the driver to the intended turn. When the intended turn is away from her/his position, it has no impact on the driver's latent lane-changing plan. When the intended turn is in a zone which is the middle of the way, the driver ignores the speed advantage opportunity. When the intended turn is close enough, the driver chooses either the appropriate or adjacent lane as maintaining or gaining speed is not important. The boundaries of the three zones, which do not depend on the driver's behavior patterns over time, are deterministic in nature. The structure of the Gipps' lane-changing model is based on his car-following model which applies some restrictions on the braking rate by drivers (Gipps, 1981). His car-following model ensures that the follower driver selects his/her speed to bring the vehicle to a safe

stop in case of a sudden stop. In Gipps lane-changing model, the deceleration of the subject vehicle is used to evaluate the feasibility to change lanes. A special braking rate is assigned to the subject vehicle so that the maximum deceleration can be achieved to complete a successful lane-changing maneuver. If the required deceleration for a lane-changing maneuver is not within the acceptance range, then this lane-changing maneuver is determined as infeasible. According to Gipps' lane-changing model, the subject vehicle driver can alter the braking rate parameter depending on the urgency of the lane-changing maneuver.

Gipps' model summarizes lane-changing process as a decision tree with a series of fixed conditions typically encountered on urban arterial and the final output of this rule-based triggered event is a binary choice (i.e., change/not change). Any new or special lane-changing reasons can be added or replaced because of its flexible structure. However, the variability in individual driver behavior is not incorporated in this model, especially the different interaction strategies among the surrounding vehicles and the subject vehicle under various traffic conditions. For example, under congested traffic conditions, either the lag vehicle gives permission to the subject vehicle to change lane, or the subject vehicle forces its way into the target lane. Although the Gipps model is used in several microscopic traffic simulation tools, it is based upon some tactically simplified assumptions and does not include any framework for model validation based on microscopic driver behavior and traffic data.

### 2.3.2.2 CORSIM Model

Halati et al. developed a lane-changing model that was implemented in CORridor SIMulation (CORSIM), in which lane changes are classified as Mandatory Lane-Changing (MLC), Discretionary Lane-Changing (DLC) and Random Lane-Changing (RLC) (Halati et al., 1997). MLC occurs when drivers merge onto a freeway or move to the target lane to make an intended turn or avoid obstructions (e.g., lane blockage, lane drop) in a lane. DLC is applied when lane changes are required for speed advantage. For instance, a driver may want to pass a slow-moving vehicle by changing to the left lane. RLC is applied when there is no apparent reason. RLC may or may not result in an advantage for the subject vehicle over its current position. In CORSIM, a certain percentage (the default value is 1%) of drivers are randomly selected to perform RLC. In this model, motivation, advantage, and urgency are considered as the three major factors behind a lane-changing decision. The motivation to change lanes depends upon either the lead vehicle speed or the lead headway threshold. The advantage factor captures the benefits of driving in the target lane. The urgency of lane-changing depends upon the number of lanes to change and the distance required to execute a complete lane-changing maneuver. In CORSIM, lane-changing maneuvers (i.e., MLC, DLC or RLC) depend on the availability of acceptable lead and lag gaps in the target lane. Acceptable lead gap is modeled utilizing the deceleration required by the subject vehicle for avoiding collision with its lead vehicle in the target lane. According to this model, the required deceleration for the subject vehicle is computed assuming the deceleration of the lead vehicle in the target lane is maximized. This computed deceleration of the subject vehicle is compared

to an acceptable deceleration which is also called the acceptable lane-changing risk. If the required deceleration is less than the acceptable risk, the lead gap is accepted and the subject vehicle initiates a lane change into the target lane.

Lane-changing algorithms used in the (FREeway SIMulation) FRESIM and (NETwork SIMulation) NETSIM are similar. The only difference lies in measuring gaps between the subject vehicle and the lead/lag vehicles in the target lane. NETSIM measures the gaps in terms of time differences, and the gaps in FRESIM are a function of time headways and speed differences. Only the FRESIM discretionary lane-changing procedure is described here. It is based on the PITT's car-following model developed by the University of Pittsburgh (Holm and Tomich, 2007). The FRESIM model assumes that the follow vehicle tries to keep a suitable gap between itself and the lead vehicle. A lane change occurs, when the follow vehicle cannot maintain the required space headway. Also in FRESIM model, an "intolerable" speed is calculated using the desired free-flow speed. The subject vehicle is eligible for a lane change, if its current speed is less than the free-flow speed. The subject vehicle driver performs a lane-changing maneuver, if her/his current speed is less than the intolerable speed.

In FRESIM discretionary lane-changing procedure, lane-changing benefits are referred to as "Advantage". Advantage is modeled through either the "lead factor" or "putative factor". The disadvantage of staying in the current lane is represented by the lead factor. On the other hand, the putative factor represents the benefits of executing lane changes. Theoretically, a subject vehicle driver could select any one of the adjacent lanes (left/right) as the target lane for performing lane changes. Thus, the advantage is

calculated for both adjacent lanes through putative factor. Based on the larger putative factor, the target lane is chosen from the adjacent lanes (left/right). Putative factor can also be determined as lead factor using putative lead headway in the adjacent lane. The overall advantage for discretionary lane change is represented by the difference between the putative factor and lead factor. It is then compared with a threshold value of 0.4 (Holm and Tomich, 2007). If the overall advantage is greater than the threshold value, a lane change occurs. So far, only the FRESIM discretionary lane-changing model has been discussed. The RLC and MLC are also incorporated in FRESIM. More detailed information on these lane-changing models could be found in (Holm and Tomich, 2007).

Additionally, after the subject vehicle moves into the target lane, a “shadow vehicle” in CORSIM is generated in the current lane in place of the subject vehicle for a while to avoid rapid speed changes of its follower. Another nice feature of CORSIM is the flexibility of taking user-provided parameters. As all drivers in CORSIM are assumed to have similar gap acceptance behavior, it does not consider the variability in gap acceptance behavior.

#### 2.3.2.3 ARTEMiS Model

ARTEMiS, which is an abbreviation for Analysis of Road Traffic and Evaluation by Micro-Simulation, is a microscopic traffic simulation model developed by Hidas (Hidas, 2005). Previously named SITRAS (Simulation of Intelligent TRAnsport Systems), this model describes lane-changing maneuvers based upon the courtesy of the lag vehicle in the destination lane (Hidas and Behabahanizadeh, 1995). In this model, a lane change is triggered by required downstream turning movements, lane drops, lane

blockages, lane use restrictions, speed advantages, or queue advantages. MLC occurs in the case of downstream turning movements, lane drops, and lane blockages and DLC happens in the early and middle distance zones. The boundaries of different zones are defined in the same way as Gipps model (Gipps, 1986). Hidas modeled each vehicle as a driver-vehicle object (DVO) using an autonomous agent technique to describe drivers' interactions involved in a complex decision-making process (Hidas, 2002). DVOs can act as giving way, slowing down or not giving way based on road congestion conditions, individual driver characteristics, and the perception of a DVO in terms of whether another DVO is trying to move into its lane or not. According to this model, lane-changing reasons are evaluated and the results are classified as “essential”, “desirable” or “unnecessary”, based on which a target lane is chosen.

In ARTEMiS, gap acceptance model selection depends on lane-changing modes. Two lane-changing modes are proposed according to traffic conditions and the necessity of changing lanes: normal lane-changing and courtesy/forced lane-changing. A normal lane change occurs when a sufficient gap is available in the target lane. This lane-changing mode is based on the Hidas car-following model and can be expressed as: a) acceptable deceleration (or acceleration) is required for the subject vehicle to follow the lead vehicle in target lane (Hidas and Behabhanizadeh, 1998), and b) acceptable deceleration is required for the lag vehicle in target lane so that the subject vehicle can safely serve as its lead vehicle.

For the courtesy/forced lane-changing mode, the subject vehicle sends a “courtesy” signal to vehicles in the target lane. Starting from the first lag vehicle, the



required deceleration is calculated using the above Hidas car-following model to allow the subject vehicle to safely merge. Based on the calculated decelerations, a follow vehicle in the target lane can be found, the new lead vehicle (to the subject vehicle) is the one right in front of the follower. A sufficient gap is created for the subject vehicle by applying the Hidas car-following algorithm to the new lead vehicle, the subject vehicle and new lag vehicle so that the subject vehicle can change lane to the target lane.

Later, Hidas categorized lane-changing maneuvers into three classes: free, forced and cooperative lane changes (Hidas, 2005). Lane-changing feasibility is checked using acceptable gaps (lead/lag). The lead and lag gaps are calculated, based on the statuses of the vehicles involved, before lane change happens. A free lane-changing maneuver is feasible, if both lead and lag gaps are greater than the desired critical gaps. If the previous condition is not satisfied, a lane change is considered “essential” and the feasibility of cooperative (courtesy) or forced lane change needs to be checked. The cooperative lane change depends on the willingness of the lag driver and the feasibility of the lane-changing maneuver. If a lag vehicle selects a certain maximum speed decrease, it indicates the willingness, which is a function of a vehicle’s aggressiveness parameter and the urgency of lane change. The lag gap at the end of deceleration can be calculated by setting the deceleration period. This represents the smallest gap between the subject vehicle and the lag vehicle after changing lanes. A cooperative lane change is feasible, if the lag gap at the end of deceleration is larger than the minimum acceptable lag gap. The forced lane change is similar to the cooperative one. The difference lies only in that the

maximum speed decrease and deceleration are assumed by the subject vehicle as average values.

Hidas validated the lane-changing model using vehicle trajectory data collected from Sydney central business district, Australia (Hidas, 2005). A total of four hours of video recording was collected from a road section where lane-changing or merging maneuvers occurred. Hidas found ambiguity between forced and cooperative lane changes by only using the trajectories from the video data. He concluded that empirical method could be designed to collect lane-changing data. One disadvantage of this model is that the given lane-changing reason set is incomplete. Lane-changing reasons, such as giving way to a merging vehicle and avoiding heavy vehicle influence, were not considered. Another downside of this model is that there is no framework for calibrating model parameters. Also, ARTEMiS is unable to resolve the conflict when a driver desires to move in one direction (left/right) for an intended turning movement and at the same time another direction to get speed advantage. Moreover, cooperative lane change and forced lane change were considered separately in this model (Hidas, 2005). However, only the lag vehicle has the ability to initiate a cooperative lane change.

#### 2.3.2.4 Cellular Automata Model

In the generic multi-lane Cellular Automata (CA) model, it is assumed that a vehicle changes to another lane if the following set of conditions is satisfied (Rickert et al., 1996):

Condition 1:  $\text{gap}_n(t) < \min(V_n(t) + 1, V_{\max})$

Condition 2:  $\text{gap}_{n,o}(t) > \min(V_n(t) + 1, V_{\max})$

Condition 3:  $\text{gap}_{n,ob}(t) > V_{\max}$

Where,

$\text{gap}_n(t)$       Number of empty cells ahead in the same lane

$\text{gap}_{n,o}(t)$       Number of empty cells ahead in the other lane

$\text{gap}_{n,ob}(t)$       Number of empty cells backward in the other lane

$V_n(t)$       Speed of vehicle n at time t

$V_{\max}$       Maximum speed of vehicles allowed

The first two inequalities or conditions above check the current and target lanes for favorable speed conditions. Then, the availability of sufficient space to perform the lane change is checked by the third condition. The lane change potential is expressed with certain probability depending on the three condition checking results. Lane-changing conditions in this model are classified as either symmetric or asymmetric. Based on this model, Nagel later proposed various additional lane-changing rules and described their characteristics in details (Nagel et al., 1998b).

#### 2.3.2.5 Game Theory Model

The game theory model is based on the giveaway behavior in a merging situation when a traffic conflict arises between through and merging vehicles, in which they try to influence each other. Kita modeled this situation based upon the game theory and

specified the game type, the number of players, and the repetition of games (Kita, 1999). He also considered the cooperative nature of the game.

First, two players are defined in the game theory lane-changing model: the merging vehicle and the through vehicle. Kita only considered two players because of the close interaction between them and neglected their interaction with the surrounding vehicles. Another key characteristic of the game theory model is the number of games to be repeated which can be one of the following three cases: each through vehicle in a conflict area plays several games; each through vehicle plays one game in a conflict area; and each merging vehicle and all through vehicles having a possible conflict with it play one game together, known as a one shot game.

It is assumed that the games are independent; and strategies of each player (i.e., the pay-off matrices) are known by the other player and non-cooperative because both players have information of each other. These two players play two different strategies: “merge” and “pass” strategies for the merging vehicle and “giveway” and “do not giveway” strategies for the through vehicle. If the merging and the through vehicles are denoted by player 1 (X1) and player 2 (X2), respectively, the pure strategy of X1, **m**, is,

$$\mathbf{m} = \{1: \text{merge}, 2: \text{pass}\}$$

And the strategy of X2, **n**, is,

$$\mathbf{n} = \{I: \text{giveway}, II: \text{do not giveway}\}$$

A pay-off matrix is developed for each player as shown in Figure 2, in which each element (i.e.,  $p_{ij}$ ,  $q_{ij}$ ) expresses the combination of situations of each vehicle.

$$\begin{array}{cc}
\begin{array}{c} \text{[X1]} \\ 1 \\ 2 \end{array} & \begin{array}{cc} \text{I} & \text{II} \\ \left\{ \begin{array}{cc} p_{11} & p_{12} \\ p_{21} & p_{22} \end{array} \right\} \end{array} & & \begin{array}{c} \text{[X2]} \\ 1 \\ 2 \end{array} & \begin{array}{cc} \text{I} & \text{II} \\ \left\{ \begin{array}{cc} q_{11} & q_{12} \\ q_{21} & q_{22} \end{array} \right\} \end{array}
\end{array}$$

Figure 2.2: Pay-off Matrices for each Player

Whether a merging car merges or a through car gives way depends on the given situation with a certain probability. Both players use mixed strategies for this type of situation. For a mixed strategy game, a bi-matrix provides at least one equilibrium solution (Aumann, 1989). Kita (Kita, 1993) modeled on-ramp merging behavior using a discrete choice model and the probability of giving way is estimated based on this game theory model. In Kita's model, drivers compare the utilities of the current lane and the target lanes (left/right) and choose the target lane with a higher utility. In this case, the utilities perceived by the drivers captured the pay-off of the players.

The maximum likelihood method is used to estimate the merging probability of the merging vehicle and the giveaway probability of the through vehicle. The estimated parameters of this model are reasonable as suggested by the likelihood ratio (0.347) and the value of the corresponding correlation coefficient (0.7) (Domencich and McFadden, 1987), showing that the game theory model is capable of explaining the real-world merging and giveaway behaviors. For congested traffic conditions, Pei and Xu developed another lane-changing model based on game theory for two types of lane-changing maneuvers (Pei and Xu, 2006). Traffic information and experience was the basis of their model to describe lane-changing maneuvers. In their model, cooperative and forced lane changes were also defined. The value of time and safety were the main factors affecting

driver behavior. When drivers are in safe situations, they will execute a lane-changing maneuver. The game theory model is largely limited to describing the merging-giveway behavior in freeway merging areas, and cannot be easily extended to other lane-changing maneuvers.

### 2.3.3 Discrete Choice-based Models

#### 2.3.3.1 Ahmed's Model

Ahmed (Ahmed et al., 1996 and Ahmed, 1999) proposed a dynamic discrete choice model to capture the heterogeneity in driving characteristics across the driving population and considered explanatory variables that affect driver behaviors. He modeled lane-changing decisions as a three-stage process: whether or not to make a lane change, target lane choice and acceptance of a gap that is sufficient to execute the lane-changing. In addition, he proposed three categories of lane-changing maneuvers: MLC, DLC and Forced Merging (FM). MLC situations apply when a driver is forced to change the current lane. DLC occurs when the driver is unsatisfied with the driving situation in the current lane and wishes to gain some speed advantage (Yang and Koutsopoulos, 1996). FM occurs when a gap is not sufficient but is created by the driver to execute a lane-changing maneuver in heavily congested traffic conditions. According to Ahmed's lane-changing model classification, lane-changing behavior is either MLC or DLC which prohibits considering any trade-offs between them. The mathematical formulation of the discrete choice framework is shown in the following functions, which describe the probability that driver  $n$  performs MLC, DLC or FM at time  $t$ :

$$P_t(LC | v_n) = \frac{1}{1 + \exp(-X_n^{LC}(t)\beta^{LC} - \alpha^{LC}v_n)}$$

LC = MLC, DLC, FM

Where,

$P_t(LC | v_n)$  The probability of executing MLC, DLC or FM for driver  $n$  at time  $t$

$X_n^{LC}$  The vector of explanatory variables affecting decision to lane changes

$\beta^{LC}$  The corresponding vector of parameters

$v_n$  The driver specific random term

$\alpha^{LC}$  The parameter of  $v_n$

In Ahmed's gap acceptance model, he defined the critical lead and lag gaps as the minimum acceptable gaps. In this model, a lane change is performed when the available lead and lag gaps in the target lane are greater than their critical gaps. The following equation represents the critical lead and lag gaps for lane-changing maneuvers of driver  $n$  at time  $t$ .

$$G_n^{cr, gap j}(t) = \exp\left(X_n^{cr, gap j}(t)\beta^{gap j} + \alpha^{gap j}v_n + \varepsilon_n^{gap j}(t)\right)$$

gap  $j$  = lead, lag

Where,

$G_n^{cr, gap j}(t)$  The critical lead and lag gaps for driver  $n$  at time  $t$

$X_n^{cr, gap j}(t)$  The vector of explanatory variables affecting the critical gap  $j$

$\beta^{gap j}$  The corresponding vector of parameters

$v_n$  The driver specific random term

$\alpha^{gap j}$  The parameter of  $v_n$

$\varepsilon_n^{gap j}(t)$   $N(0, \sigma_{\varepsilon j}^2)$  is a random term

The probability of accepting a gap during a MLC, DLC or FM for driver  $n$  at time  $t$  is given as follows:

$$\begin{aligned} P_n(\text{gap acceptance} | v_n) &= P_n(\text{lead gap acceptable} | v_n) \times P_n(\text{lag gap acceptable} | v_n) \\ &= P_n(G_n^{lead}(t) > G_n^{cr, lead}(t) | v_n) \times P_n(G_n^{lag}(t) > G_n^{cr, lag}(t) | v_n) \end{aligned}$$

Where,

$G_n^{lead}(t)$  The probable lead gaps in the target lane

$G_n^{lag}(t)$  The probable lag gaps in the target lane

Ahmed subsequently implemented his model in MITSIM (Microscopic Traffic Simulator). MITSIM was developed primarily to assess Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS) at the operational level. Although his lane-changing model was unable to capture the trade-offs between MLC and DLC decision processes, it accurately described the differences between



drivers' MLC, DLC and FM decisions. For instance, in MITSIM drivers are unable to overtake when mandatory considerations are active. Similar to the Gipps model, the existence of an MLC is determined based upon the distance of the subject vehicle to the downstream exit ramp. In addition, a dummy variable is introduced to capture the differences in acceptable gap values between a passenger car and a heavy vehicle when the heavy vehicle is the subject. Though this very coarse and simplistic method accounts for the differences in operational characteristics of these two vehicle types, the above models incorporate a rigid separation between MLC and DLC, which is unrealistic in real-life driving.

#### 2.3.3.2 Toledo et al.'s Model

Toledo et al. developed a probabilistic lane-changing decision model to describe the trade-offs between MLC and DLC (Toledo, 2002). The trade-offs between MLC and DLC are captured by considering both types of lane changes in a single utility function. A discrete choice framework is employed to model drivers' tactical and operational lane-changing decisions. The model is calibrated using the maximum likelihood estimation technique (Toledo et al., 2003). The lane-changing decision model consists of 1) choice of the destination lane, and 2) decision for accepting gap. Four groups of explanatory variables are considered in the model underlying lane-changing decisions: neighborhood variables (e.g., gaps, speeds), path plan variables (e.g., distance from the intended exit off-ramp), network knowledge and experience (e.g., avoiding the nearest lane next to the shoulder), and driving style and driving capabilities. In the target lane model, the set of target lane choices includes: 1) remaining in the current lane, 2) shifting to the right, and

3) shifting to the left adjacent lane. The target lane choice model, probability of selecting a specific lane, and critical gap model are similar to those in Ahmed's model. In this model, the decision of selecting the target gap is based on the target lane choice. The model assumes that the driver will change lane to the target lane based on the acceptance of the lead and lag gaps in the target lane and does not consider any other gaps. Toledo et al. defined the critical lead and lag gaps as the minimum acceptable gaps. When the available target lead and lag gaps are greater than their corresponding critical values, they will be accepted. A lognormal distribution is assumed for the critical gaps to ensure they are always positive.

According to this model, after selecting a target lane and finding gaps of sufficient sizes, the subject vehicle driver performs a sequence of accelerations and decelerations in order to move into the target lane (Toledo et al., 2007). Toledo et al. used a conditional probability to determine whether a lead/lag gap is acceptable or not.

In Toledo's model, the subject vehicle employs a three-stage acceleration behavior model to select the target gap. First, if the subject vehicle driver wishes to remain in the current lane, a stay-in-the-lane selection model applies. Second, if the driver accepts the available target gap and changes into an adjacent lane, an acceleration model applies for changing lane. Third, if the subject vehicle driver initially accelerates or decelerates for changing lane but later rejects the target gap, a target gap acceleration model applies.

This lane-changing model was implemented in MITSIM and tested using detailed vehicle trajectory data collected in Arlington, VA. The purpose of the implementation

was to estimate travel time, speed, and the distribution of traffic volumes across lanes. During the implementation, the MLC and DLC models were first separated and later integrated. The estimated values by MITSIM were then compared against the observed values. In the case of travel time and speed, both the separated and integrated scenarios resulted in differences between the observed and estimated values. The travel time differences of the separated and integrated scenarios were 3.20% and 9.50%, respectively. For speed, the corresponding values were -5.60% and -2.90% respectively. But, the estimated and observed distributions of traffic volumes across lanes were similar for both the separated and integrated scenarios. The main weakness of this lane-changing model is the difficulty of determining the utility functions for various decision choices. Built upon this work, Choudhury et al. proposed a cooperative and forced gap acceptance model for congested traffic conditions (Choudhury et al., 2007).

#### 2.3.4 Artificial Intelligence Models

##### 2.3.4.1 Fuzzy Logic-Based Models

Fuzzy logic-based models consider the uncertainty of lane-changing maneuvers and take into account the natural or subjective perception of real variables (Ma, 2004). The unique nature of fuzzy logic models is that they can translate nonlinear systems into IF-THEN rules (Mendel, 1995). Fuzzy-LOGic-based motorWay SIMulation (FLOWSIM) is a simulation model built upon fuzzy sets and systems (McDonald et al., 1997). In this model, lane-changing maneuvers are based on two premises, changing to a slower lane and changing to a faster lane. Das et al. proposed a new microscopic simulation methodology based on fuzzy rules for implementation in the Autonomous Agent

SIMulation Package (AASIM) software (Das and Bowles, 1999). In this fuzzy-logic based model, lane-changing maneuvers are classified as MLC and DLC. MLC fuzzy rules consider the distance to the next exit or merge point and the required number of lanes to change. DLC is a binary decision that is based on the driver's speed satisfaction (Das et al., 1999), but it does not consider vehicle types in lane-changing decisions. Moridpour et al. also developed a lane-changing model using fuzzy logic, which is used to predict the lane-changing maneuver of heavy vehicles on freeways (Moridpour et al., 2012). This model considers three types of lane-changing behavior: motivation of lane-changing, selection of the target lane and execution of the lane-changing maneuver. Because of abstract fuzzy rules and membership functions, the recalibration and validation process for fuzzy logic-based lane-changing models is fairly complex.

#### 2.3.4.2 Artificial Neural Network Model

Artificial neural network (ANN) models process information using functional architecture and mathematical models that are similar to the neuron structure of the human brain. These models learn human behaviors from training and are capable of demonstrating those human behaviors in a new situation. In recent years, neural networks have also been used for modeling driver behavior in the transportation field (Hunt and Lyons, 1994). For instance, Hunt and Lyons predicted drivers' lane-changing decisions using neural networks on dual carriageways (Hunt and Lyons, 1994). Neural network models are completely data-driven and require supervised training by field-collected traffic data before they can be used to predict driving behavior. Their dependence on the availability of field-collected traffic data is the main disadvantage of neural network

models, although previous results show that they can accurately predict lane-changing behavior (Dumbuya et al., 2009).

Dumbuya et al. developed Neural Driver Agents (NDA) for modeling lane-changing maneuvers (Dumbuya et al., 2009). A multilayer NDA model was designed and implemented. A back-propagation training algorithm was used to train the NDA model, which takes inputs such as current direction of the vehicle, current speed, the distance from vehicle, preferred speed and current lane. The output of the model includes new direction and new speed. This NDA model learned lane-changing behavior from known situations using data collected from the TRL (Transport Research Laboratory) driving simulator. The authors then used the fitted NDA model to predict driver behavior for unseen situations. They demonstrated that NDA has the ability to properly model lane-changing maneuvers. Later, the NDA model was incorporated into the commercial NeuroSolutions software package developed by NeuroDimension.

During the study using the driving simulator, Dumbuya et al. recruited eight participants to “drive” on a simulated two-lane highway. At first the participants were in lane 1. They changed to lane 2 to overtake a slow-moving vehicle and returned back to lane 1 as if they were on a real UK highway. For each completed simulation, a set of data was recorded. Using those data sets, they trained the NDA model. When the training process was completed, the trained model was used to simulate the vehicle trajectory. It was found that the simulated vehicle followed a realistic path around the lead slow-moving vehicle. This result shows the changes in direction generated by NDA model match those of real drivers when executing an overtaking maneuver at a speed of 70 mph.

The reasonably close lane-changing behaviors of humans and NDA suggest that the NDA is a promising tool to replicate a wide range of lane-changing behaviors (e.g., aggressive, tired, alcohol-impaired, learner drivers). However, the results also show that the NDA is unable to accurately model lane-changing trajectories when the travel speed is either low or high (Tomar et al., 2010).

### 2.3.5 Incentive-Based Models

#### 2.3.5.1 MOBIL

The MOBIL lane-changing model is based on two criteria: incentive and safety. The incentive criterion measures the attractiveness of a given lane based on its utility, and the safety criterion measures the risk associated with lane-changing (i.e., acceleration) (Treiber et al., 2000, and Treiber and Helbing, 2002). According to this model, the target lane is more attractive to the driver of the subject vehicle if the incentive criterion is met. A lane change takes place if the safety criterion is satisfied as well. The MOBIL rules are applied for simulation of multilane traffic in the Intelligent Driver Model (IDM) (Treiber and Kesting, 2007). In IDM, two types of passing rules are considered for lane changes: symmetric and asymmetric. The symmetric passing rules are based on safety and incentive criteria. They are applied when changing to the right lane is not strictly forbidden. When the deceleration ( $a'$ ) of the follow vehicle ( $F'$ ) in the target lane is equal to the IDM braking deceleration ( $a'_{IDM}$ ), the safety criterion is satisfied. For a lane change to happen, the deceleration of the follow vehicle should also not exceed a certain limit  $b_{safe}$  as shown below.

$$a' (F') > - b_{safe}$$

The incentive criterion is determined by weighing the lane-changing advantage against imposed disadvantage to other vehicles. The increased acceleration (or reduced braking deceleration) is the measure of advantage to the subject vehicle before and after the potential lane change. The total decreased acceleration or increased braking deceleration is the measure of disadvantage to vehicles in the target lane. In this model, the lane-changing decision is also influenced by a politeness factor  $p$ . This politeness factor  $p$  will be further described later and its value is typically less than 1.

The disadvantages of target-lane vehicles, advantage of the subject vehicle, politeness factor  $p$  all affect the lane-changing decision. Thus, typical strategic features of classical game theory have been incorporated in MOBIL (Treiber et al., 2000). It can describe different driving behaviors by varying the politeness factor ( $p$ ), while other lane-changing models typically assume the politeness factor to be zero (0). In MOBIL,  $p > 1$  is for an altruistic driving behavior;  $0 < p < 0.5$  is for a realistic driving behavior;  $p = 0$  is for a purely selfish driving behavior; and  $p < 0$  is for a malicious driving behavior.

A special case of this model is given by  $p = 1$  and lane-changing acceleration threshold,  $a_{thr} = 0$ . For this special case, a lane-changing maneuver will take place whenever the sum of the advantage and disadvantage of all affected drivers is positive after the change. This explains the acronym for this model, which is: MOBIL = Minimizing Overall Braking Decelerations Induced by Lane changes.

The asymmetric rules are applied in many European countries where changing to the right lane is prohibited, unless traffic is congested or the subject vehicle is forced to change to the right lane (i.e., on-ramp, off-ramp, lane drop). A lane-usage bias rule is introduced to capture this asymmetric situation. This rule only represents operational lane-changing decisions. However, a lane-changing model should be able to describe both strategical and tactical aspects of lane-changing behaviors for mandatory lane changes and for congested traffic conditions.

#### 2.3.5.2 LMRS

Schakel et al. proposed a LMRS (Lane-changing Model with Relaxation and Synchronization) lane-changing model, based on drivers' desire to change lanes (Schakel et al., 2012). The desire is a combination of the route, speed and keep-right incentives. A trade-off is considered within the combination of incentives with the route incentive being dominant. The following equation is a sample combination of incentives representing the desire to change from lane  $i$  to lane  $j$ :

$$d^{ij} = d_r^{ij} + \theta_v^{ij} * (d_s^{ij} + d_b^{ij})$$

Where,

$d^{ij}$  Combined desire to change lane from  $i$  to  $j$

$d_r^{ij}$  Desire to follow a route

$d_s^{ij}$  Desire to gain speed



$d_b^{ij}$	Desire to keep right
$\theta_r^{ij}$	Voluntry (discretionary) incentives

The total desire determines drivers' lane-changing behaviors. The range of meaningful desire is from -1 to 1. Negative values represent that a lane change is not desired, and positive values mean the driver wants to change lane. Depending upon the desire value, Schakel et al. further classified lane changes as Free Lane-Changing (FLC), Synchronized Lane-Changing (SLC) and Cooperative Lane-Changing (CLC).

$$0 < d_{free} < d_{sync} < d_{coop} < 1$$

Schakel et al. also considered a relaxation phenomenon in their model. As in the real world, drivers may accept small gaps for a large desire. For very small desire values, no lane changes will occur. For a relatively large desire, FLC will happen and no preparation is required. In case of SLC and CLC, the subject vehicle speed needs to be synchronized with the speeds of vehicles in the target lane for creating a gap. This behavior is also called synchronization.

The gap acceptance module in this model is similar to MOBIL. In addition, this model considers an applicable headway for gap acceptance. A gap is accepted if the accelerations of the subject vehicle and new follower is larger than a safe deceleration threshold. According to this model, large decelerations and short headways can be accepted for a large desire and the relaxation of headway values is exponential with relaxation time. The subject vehicle driver will synchronize her/his speed, if the lane-changing desire is above the synchronization threshold ( $d_{sync}$ ). She/he will synchronize

the speed with the target lane speed by applying a maximum deceleration which is both comfortable and safe. A Gap can be created, if adjacent leader lane-changing desire is above cooperation threshold.

Schakel et al. used a modified version of Intelligent Driver Model (IDM) developed by Treiber et al. [Treiber et al., 2000] to evaluate the proposed lane-changing model. They referred to this new simulation model as IDM+, based on which they calibrated and validated the LMRS model in both free-flow and congested traffic conditions. The main goal of their study was to accurately represent real-world observations at the lane level such as the lane volume distribution, lane-specific speed, and progression of congestion. Their lane-changing model has a set of seven (7) parameters with physical and intuitive meanings. The full model, combining the LMRS and IDM+, has twenty (20) parameters. Schakel et al. tried to alleviate the calibration difficulties by considering the two flow scenarios (i.e., free flow and congested) separately. They calibrated and validated the model using data from a segment of A20 freeway near Rotterdam in Netherlands. This segment included a few on- and off- ramps and a lane drop. The data was collected utilizing loop detectors which were closely spaced (300-500m). Although realistic lane volume distributions and lane-specific speeds were generated for the free-flow condition, the model fitting result for the congestion condition was unclear. Furthermore, the generalization ability of their lane-changing model is unknown for scenarios with different levels of congestion and numbers of lanes.

### 2.3.6 Theoretical Comparison of Lane-Changing Models

Based on the review of existing lane-changing models, rule-based and discrete choice-based models appear to be the most popular ones. These models have been widely implemented in microscopic traffic simulators. Among them, rule-based lane-changing models are based on the perspective of drivers. For rule-based models, typically the subject vehicle's lane-changing reasons are evaluated first. If these reasons warrant a lane change, a target lane from the adjacent lane(s) is selected. A gap acceptance model fitted based on field data/simulation data is then used to determine whether the available gaps should be accepted.

Most discrete choice-based lane-changing models are based on logit or probit models. For discrete choice-based models, the lane-changing maneuver is usually modeled as either MLC or DLC following three steps: 1) checking lane change necessity, 2) choice of target lane, and 3) gap acceptance. Each of these steps can be formulated as a probit or logit model. Depending on which step and the number of lanes, the subject driver may face a binary or multi-choice decision. Similar to rule-based models, discrete choice model parameters and utility functions need to be calibrated using field collected data. In existing discrete choice-based lane-changing models, the heterogeneities in drivers and vehicles (i.e., driver aggressiveness, driving skill level, vehicle acceleration performance) have not been given adequate consideration. . A major reason is that existing traffic data and data collection technologies cannot provide information that is detailed enough for developing and testing such models. Nevertheless, these characteristics are important for accurately describing real-world lane-changing behaviors

and relevant explanatory variables should be incorporated into the utility functions of future discrete choice-based lane-changing models.

Artificial Neural Network (ANN) lane-changing models are completely data-driven and fundamentally different from the rule-based and discrete choice-based models. Although researchers can specify some network parameters such as numbers of input units, hidden neurons, and layers, they have very little control over the model structure (such as the utility functions in discrete choice-based models). ANN models have to be trained and validated using field-collected microscopic traffic data before they can be used to predict any lane-changing behavior. The fitted ANN model parameters do not have practical meaning either and cannot be interpreted as those in discrete choice-based models. Fuzzy logic-based models describe lane-changing behaviors using fuzzy rules and membership functions. Compared to other models, a major advantage of them is that they can better incorporate human experience and reasoning into the development of lane-changing models. However, it is not an easy task to determine the fuzzy membership functions and rules. The calibration process of Fuzzy logic-based models is very difficult.

The idea behind the incentive-based models is intuitive and straightforward: drivers choose to change or not change lanes in order to maximize their benefits. It is similar to the utility function concept in discrete choice-based models. However, there are multiple utility functions in a discrete choice-based model and the value of each utility function represents the utility (or “advantage”) of a choice alternative. In incentive-based models such as MOBIL, there is only one “advantage” value, which is compared against a threshold value for final decision making. An advantage of the incentive-based model

LMRS is that it takes into account driver's desire to follow a route into consideration. This may potentially generate more realistic lane-changing behaviors. For instance, through traffic drivers on a multilane highway typically tend to stay away from the rightmost lane to avoid the interference of exiting and entering traffic. This model also has a flexible structure and additional incentives may be easily integrated into it. The above discussions provide a brief summary and theoretical comparison of the reviewed lane-changing models. A more detailed and systematic comparison of the four groups of models is presented in Table 2.1.

Table 2.1: Theoretical Comparison of Lane-Changing Model Categories

Microscopic Lane-Changing model			
Rule-based Model	Discrete Choice-based Model	Artificial Intelligence Model	Incentive-based model
Lane-Changing Decision			
Decide on decision tree with series of fixed condition	Utilize logit or probit model	Based on driver-vehicle status	Decide on lane-change desire (LMRS)
Reason for Lane-Changing			
Decide whether lane-changing applies or not through explanatory Variables (EV) EV: Maximum subject vehicle's safe speed and brake, front gap, subject vehicle driver's estimation of front vehicle driver's brake	Explanatory Variable for gained utilities are: MLC-Exit/merge distance, number of lane changes, DLC-Presence of heavy vehicle, front relative speed and deceleration	<ul style="list-style-type: none"> <li>• Completely data driven and require supervised training</li> <li>• Fuzzy sets and systems</li> </ul> EV: MLC-Exit/merge distance, number of lane changes, DLC-Left and right lane density, drivers' satisfaction	Measure level of lane-changing desire based on speed incentive, Route incentive, Keep right incentive
Target lane selection			
Decide on fixed lane-changing purpose or advantage for lane-changing EV: Acceptable lead and lag gaps, Critical gaps	At each stage, utilities for all alternatives are calculated in the lane-changing process EV: Target lead and lag gaps and relative speeds, subject vehicle speed, presence of heavy vehicle, tailgating, avoiding the rightmost-lane, distance to the exit off-ramp	Fuzzy rules, Drivers' recent speed history, and the level of congestion <ul style="list-style-type: none"> <li>• Change lanes to left or right</li> </ul> EV: Left-Motivation, opportunity, Right-Pressure, Gap satisfaction	Depend on level of lane-changing desire EV: Anticipation Speed, Maximum vehicle speed, Desired speed, Anticipation distance, Speed limit, Speed gain
Gap acceptance			
Gap acceptance parameters for are picked up from field/simulation data, and calculated using gap acceptance formulae	<ul style="list-style-type: none"> <li>• Permission of lane change decides on the lead and lag gap acceptance</li> <li>• Gap acceptance</li> </ul> EV: Target lead and lag relative speeds, distance between target lead and lag	Consider the safe headway to the front vehicle in the current lane <ul style="list-style-type: none"> <li>• Find a gap in target lane</li> </ul> EV: Front, lead and lag gaps and relative speeds. <ul style="list-style-type: none"> <li>• Accept sufficient size gap</li> </ul> EV: Target lead and lag speeds and gaps, exit/merge distance	Based on deceleration rate utilizing the car-following model <ul style="list-style-type: none"> <li>• Find a gap in target lane</li> </ul> EV: Front, lead and lag gaps and relative speeds <ul style="list-style-type: none"> <li>• Accept sufficient size gap</li> </ul> EV: Target lead and lag acceleration and time headway, deceleration threshold

Table 2.1: Theoretical Comparison of Lane-Changing Model Categories (Continued)

Microscopic Lane-Changing model			
Rule-based Model	Discrete Choice-based Model	Artificial Intelligence Model	Incentive-based model
<b>Divers variability</b>			
Does not consider driver's variability on gap acceptance	Does not consider invariant characteristics of drivers and their vehicles for a given driver over time and choice dimensions such as choice of target lane, gap acceptance	Attempt to capture drivers' variability with training data sets of driver behaviors	Capture driver's variability using politeness factor (MOBIL) and accepted headway, deceleration, and level of desire (LMRS)
<b>Advantages</b>			
<ul style="list-style-type: none"> <li>• Simplicity in modeling</li> <li>• Decision process in one simple stage, Small number of variables</li> </ul>	<ul style="list-style-type: none"> <li>• Decide on the basis of maximum gained utility</li> <li>• Probabilistic results instead of binary answers (yes/no)</li> </ul>	Consider human's imprecise perception, require numerical data, calibrating using optimization algorithm	<ul style="list-style-type: none"> <li>• Small number parameters</li> <li>• Take into account drivers variability</li> </ul>
<b>Disadvantages</b>			
<ul style="list-style-type: none"> <li>• Difficulties in calibrating the model parameters.</li> <li>• Use only primary variables</li> <li>• Binary answers (yes/no)</li> </ul>	<ul style="list-style-type: none"> <li>• Require to calculate probability functions to determine the utility of each choice</li> </ul>	<ul style="list-style-type: none"> <li>• Difficulties and complexity in fuzzy rules, membership functions</li> <li>• Require large amount of data</li> </ul>	<ul style="list-style-type: none"> <li>• Fit in congestion is unclear</li> <li>• MOBIL only considers operational process</li> </ul>
<b>Applications</b>			
These models are utilized in microscopic traffic simulators and are applicable to capacity analysis.			

## **2.4 Summary of Literature Review**

Various car-following models and lane-changing models were discussed in this chapter. Lane-changing models are out of the scope, in terms of analysis, for this thesis. However, these models and car-following models collaboratively represent any traffic stream. The Gazis-Herman-Rothery (GHR) model is one of the oldest and well developed car-following models. However, the model suffers from the issue that its parameters vary in different driving conditions. The Linear model has been studied extensively, similar to GHR model; although it has relatively a simple and linear form, the difficulty in the parameter calibration makes it less popular. Due to the nature of car-following behaviors, applying fuzzy logic in the car-following theory seems to be a reasonable attempt. However, the difficulty in calibrating the membership function, the key concept in fuzzy logic, limits the application of such attempts. On the other hand, Gipps' variation of the Collision Avoidance (CA) model is probably the most widely used car-following model for simulation purposes. The Optimal Velocity (OV) model is a relative new car-following model, which was first proposed in 1990. The model is unique in presenting stop and-go and congested traffic conditions. Two variances of OV model were proposed later to improve the issue of OV model including the data agreement and the startup process. Several studies attempted to calibrate car-following models with different data sets which is described in chapter one. Most of the application of such methodology indicated the efficacy of the methodology on the macroscopic level; however the performance in the microscopic level is unclear.



## CHAPTER THREE

### CALIBRATION METHOD

#### 3.1 Introduction

In this chapter, the method employed for calibrating car-following models is discussed. This calibration approach is based on Bayesian estimation theory and it utilizes Markov Chain Monte Carlo (MCMC) simulation. Figure 3.1 shows the general steps for analysis of a car-following model.

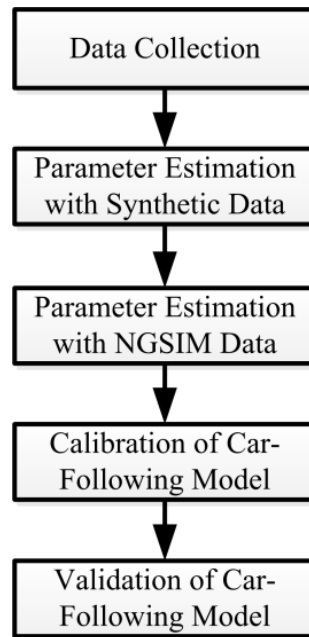


Figure 3.1: Calibration and Validation Steps of a Car-Following Model

#### 3.2 General Calibration Method

A method is developed for parameter estimation and calibration of car-following models which is based on Markov Chain Monte Carlo (MCMC) method using Bayesian

estimation theory that have been recently investigated for inverse problem. The Bayesian framework used prior distributions and vehicle trajectory data to estimate the statistical distribution of the parameters of car-following models. The general Bayesian Framework for Calibration of a Car-Following Model is illustrated in Figure 3.2. In this framework, the prior probabilities are transformed into posterior probabilities for each parameter of the car-following model, for which Bayes' rule is used. After that, the Metropolis Hasting algorithm is used to calculate the Bayes estimate of car-following model parameters. Finally, another real world dataset is utilized to validate the car-following model.

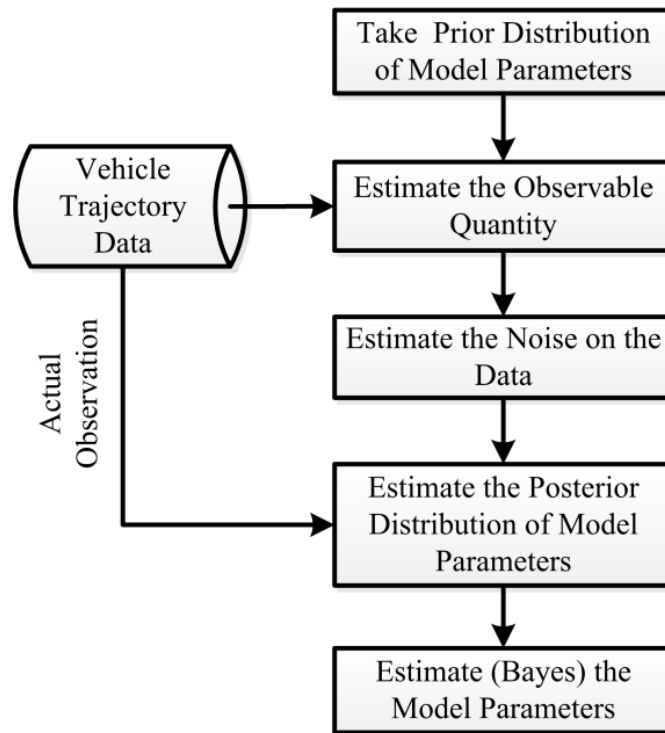


Figure 3.2: Bayesian Framework for Calibration of a Car-Following Model

### 3.2.1 Statistical Inverse Problem

Bayesian statistics provide a theory of inference which enables the creation of a relationship between the results of observation with theoretical predictions. Consider a parameter vector,  $v(q)$  and the result of the observations represented by an observation vector,  $m(q)$ . Figure 3.3 shows the definition of an inverse problem. Let  $d$  be the actual observation of  $m$  mainly  $m + \varepsilon$  (noise).  $P(d|q)$  is the conditional probability of the observation given the cause. On the other hand,  $P(q|d)$  which is the conditional probability of the possible causes, given that some effect has been observed. This inverse probability represents our state of knowledge of  $v$  after measuring  $m$ . In the context of inverse problem theory,  $v(q)$  is the image and  $m(q)$  is the data.

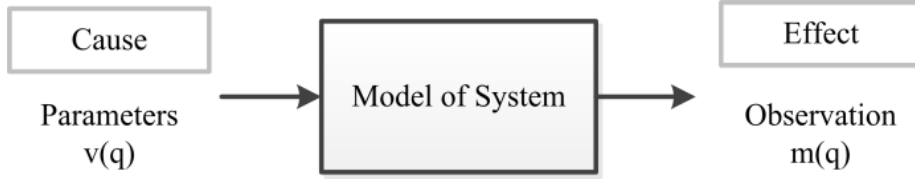


Figure 3.3: Definition of Inverse Problem

### 3.2.2 Bayesian Inference: From Prior to Posterior

In a Bayesian framework, the prior distribution of the parameter sets of the car-following model is used to find the posterior distribution of the parameter while utilizing Bayesian inference. Let  $\mathbf{q}$  be the vector of model parameters,  $\mathbf{q} = (q_1, q_2, q_3, \dots, q_k)^T$  with  $k$  elements of a given car-following model. Consider a generic model  $v(\mathbf{q})$  and then a model of the observation using  $m(\mathbf{q}) := C[v(\mathbf{q})]$ , where  $C$  transforms velocity to an

observable quantity such as acceleration, head way, etc. Let  $d$  be the actual observation of  $m$  that is,

$$d = m + \varepsilon(\text{noise})$$

Now consider the Bayes formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

$P(A)$  = The prior degree of belief in  $A$

$P(A|B)$  = The posterior degree of belief having accounted for  $B$

$\frac{P(B|A)}{P(B)}$  = The support  $B$  provides for  $A$

From the above Bayesian inference, the posterior probability distribution of  $\mathbf{q}$  can be easily obtained by the following equation:

$$P(\mathbf{q}|d) = \frac{P(d|\mathbf{q})P(\mathbf{q})}{P(\mathbf{q})} = \frac{P(d|\mathbf{q})P(\mathbf{q})}{\int_{\mathbb{R}^k} P(d|\mathbf{q})P(\mathbf{q})d\mathbf{q}} = \frac{P(d|\mathbf{q})P(\mathbf{q})}{c}$$

where,  $P(\mathbf{q})$  represents the prior distribution of  $\mathbf{q}$ , which is an initial guess on how  $\mathbf{q}$  should be distributed. It is worthy to note that this is similar to a regularization term using a deterministic method. For example,  $P(\mathbf{q})$  may be guessed from prior studies about the distribution of a particular parameter in question. The next step is to estimate the distribution of  $P(d|\mathbf{q})$ , which is called the Bayes estimate of the parameter.

### 3.2.3 Bayesian Estimate

In the following, the prior distribution of the parameter of the car-following model is assumed to be a multivariate normal, which means that the mean square error is to be minimized, that is:

$$P(d|\mathbf{q}) \propto e^{-\frac{1}{2}(\mathbf{d}-\mathbf{m}(\mathbf{q}))^T \Sigma^{-1}(\mathbf{d}-\mathbf{m}(\mathbf{q}))}$$

where  $\Sigma$  is the corresponding covariance matrix. Thus, the Bayes estimate is obtained, which is the expected value from  $\mathbf{q}$  given by the following definition:

$$E(\mathbf{q}|d) := \int_{\mathbb{R}^k} \mathbf{q} P(\mathbf{q}) d\mathbf{q}$$

This integral is difficult to solve. The Markov chain Monte Carlo (MCMC) method can be used to solve this integral.

### 3.2.4 Markov Chain Monte Carlo (MCMC) Method

The MCMC method is used to solve the integral. A large number of random samples are needed from the posterior distribution of  $\mathbf{q}$  for the Bayes estimate. The Gibbs sampler and the metropolis Hasting algorithm are the typical algorithms, which are used to generate such large number of random samples. In this case study, a special type of the Metropolis Hasting algorithm (Figure 3.4) is used. For a more general form, see reference (Gelman et al., 2004).

Consider that a large number of random samples  $(\mathbf{q}^{(i)})_{i=1}^r$  are generated, with  $r$  random samples after the burn-in time, from the posterior distribution of  $\mathbf{q}$  and then approximate the Bayesian estimate by its sample mean,

$$E(\mathbf{q}|d) \approx \frac{\sum_{i=1}^r \mathbf{q}^{(i)}}{r}$$

Select a large enough  $r$ , which is the desired quantity of random samples from the posterior distributions, and a large enough burn-in time (b). Now given a current sample  $\mathbf{q}^{(t)}$  we generate a new random sample  $\mathbf{q}^{(t+1)}$  using the following algorithm:

Step 1: Generate  $\tilde{\mathbf{q}} \sim N(\mathbf{q}^{(t)}, \Psi)$ , where  $\Psi$  is a covariance matrix.

Step 2: Calculate the acceptance ratio:

$$a := \frac{P(\tilde{\mathbf{q}}|d)}{P(\mathbf{q}^{(t)}|d)} = \frac{\frac{P(d|\tilde{\mathbf{q}})P(\tilde{\mathbf{q}})}{c}}{\frac{P(d|\mathbf{q}^{(t)})P(\mathbf{q}^{(t)})}{c}} = \frac{P(d|\tilde{\mathbf{q}})P(\tilde{\mathbf{q}})}{P(d|\mathbf{q}^{(t)})P(\mathbf{q}^{(t)})}$$

Step 3: If,  $a \geq 1$ , set  $\mathbf{q}^{(t+1)} = \tilde{\mathbf{q}}$ . Else set  $\mathbf{q}^{(t+1)} = \tilde{\mathbf{q}}$  with probability  $a$ , and  $\mathbf{q}^{(t+1)} = \mathbf{q}^{(t)}$  with probability  $1-a$ .

Step 4: Stop if  $r+b$  samples are produced, otherwise set  $t = t + 1$  and go to step 1.

In general, any parameter set can be used as the starting element  $\mathbf{q}^{(0)}$ . However,  $\mathbf{q}^{(0)}$  may be selected from calibration results from previous studies so that the burn-in time

could be minimized. The burn-in time can be defined as the length of time that one spends to let the Metropolis Hasting algorithm run before starting to collect actual samples of the parameter. This is important as running this algorithm can be very time-consuming.

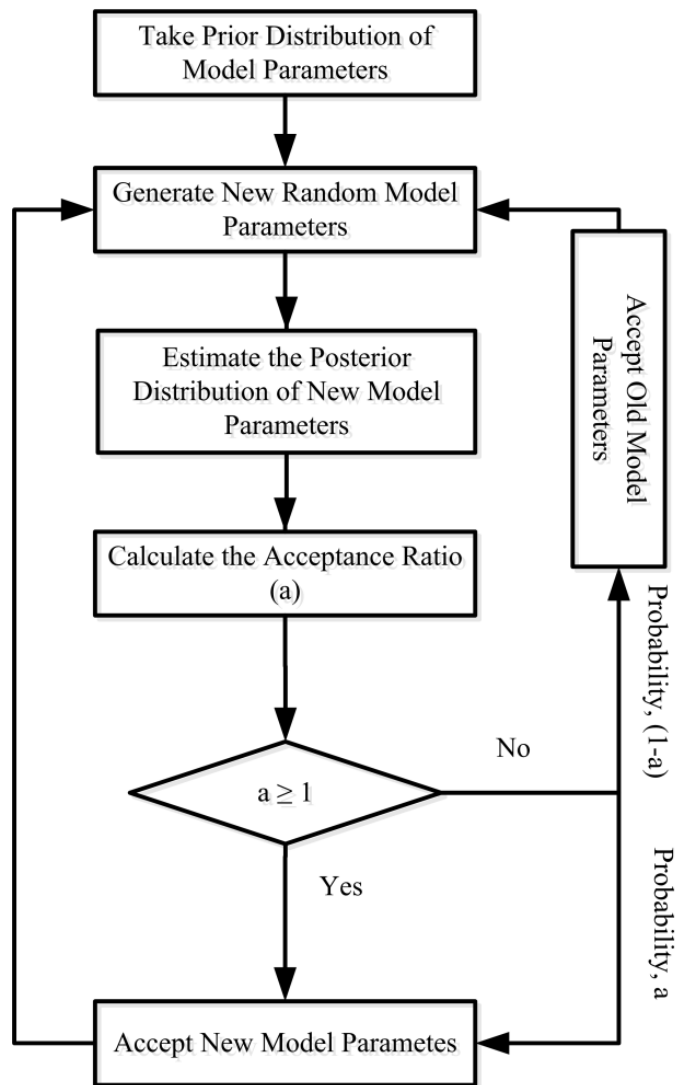


Figure 3.4: Bayesian estimation process using Markov Chain Monte Carlo (MCMC) method (Metropolis-Hasting algorithm)

### **3.3 General Framework for Validation of the Calibration Method**

The following steps can be taken to validate a calibration method of car following models. Synthetic data can be generated using known distributions of a parameter set of a car-following model. Then, the known distributions of the parameter set can be compared with the distributions of the estimated corresponding parameter set generated by calibrating the car following model through the calibration method that is being validated and the generated synthetic data. Figure 3.5 shows the general step-by-step process for the validation of stochastic calibration method. In order to generate the distribution of the parameter set, we need to calculate the mean and standard deviation which we get from the previous study. Using the mean and standard deviation, the normal distribution of parameter set can be generated for a specific number of vehicles. After that each parameter set could be assigned to each vehicle and parameter set must be constant for all observations for a specific vehicle. Then the observable quantity of any car-following model can be calculated using the parameter set and vehicle trajectory data set. Finally the distribution of calibrated parameters using the Bayesian framework is compared with the generated distribution of parameters from the given mean and standard deviation. If both of that distribution matches each other, the validation of the calibration method is completed.



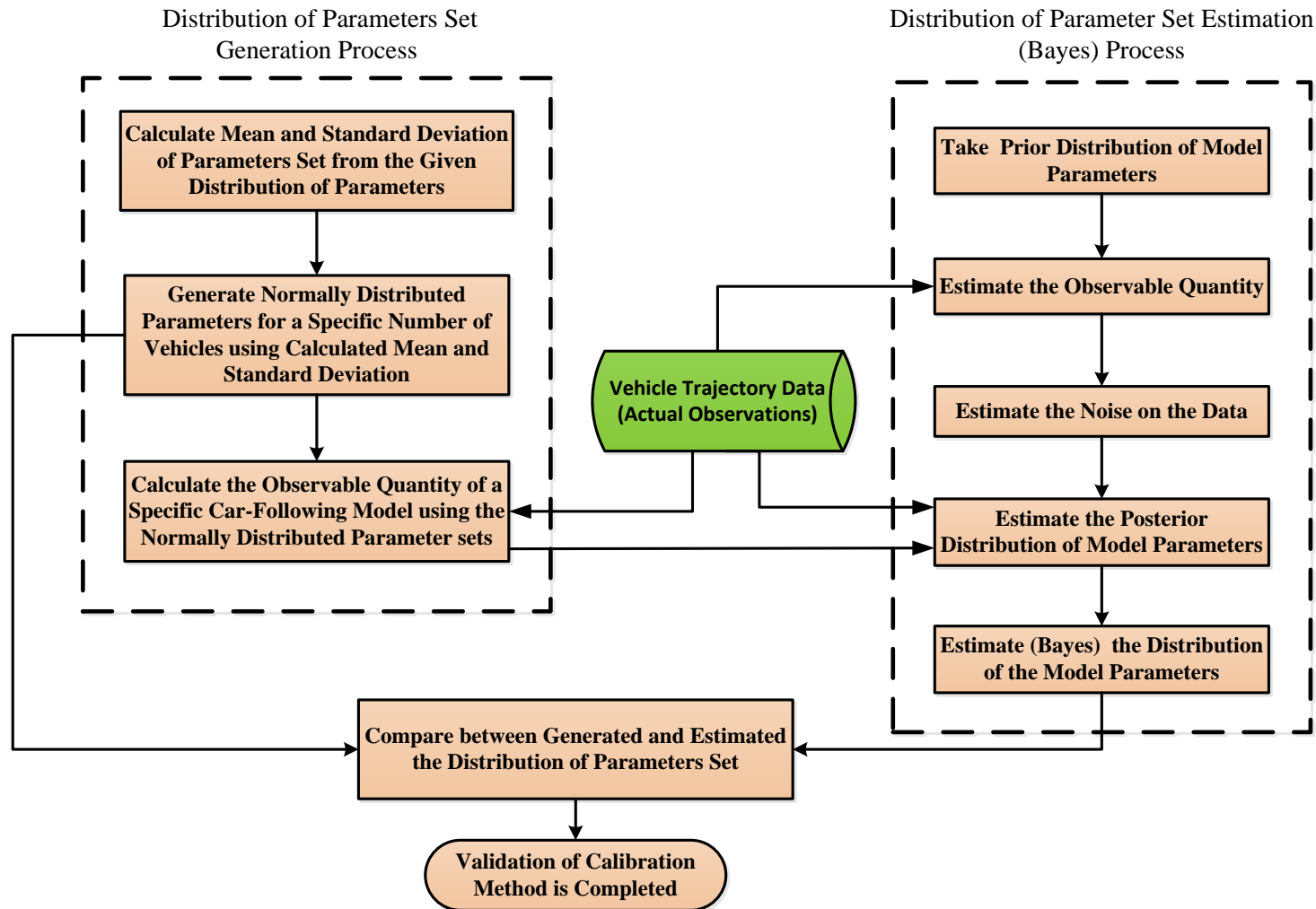


Figure 3.5: Validation of the Bayesian Framework Calibration Method

## CHAPTER FOUR

### MODEL CALIBRATION AND VALIDATION RESULTS

#### 4.1 Introduction

This chapter describes vehicle trajectory dataset utilized in this study. The results from parameter estimations and validation of linear car-following model are also presented in this chapter.

#### 4.2 Dataset description

The data set representing 45 minutes of data collected during the afternoon peak period on a segment of Interstate 80 in Emeryville (San Francisco), California. The data set consists of detailed vehicle trajectory data, wide-area detector data and supporting data needed for behavioral research.

##### 4.2.1 Data Collection Procedure

Data used in this thesis represent travel on the northbound direction of Interstate 80 in Emeryville, California. This data was collected using video cameras mounted on a 30-story building, Pacific Park Plaza, which is located in 6363 Christie Avenue and is adjacent to the interstate freeway I-80. The University of California at Berkeley maintains traffic surveillance capabilities at the building and the segment is known as the Berkeley Highway Laboratory (BHL) site. Video data were collected using seven video cameras.

#### 4.2.2 Study Area Description

Figure 4.1 provides a schematic illustration of the location for the vehicle trajectory dataset. The site was approximately 1650 feet in length, with an on-ramp at Powell Street. The off-ramp at Ashby Avenue is just downstream of the study area. Lane numbering is incremented from the left-most (the high-occupancy vehicle (HOV) lane).

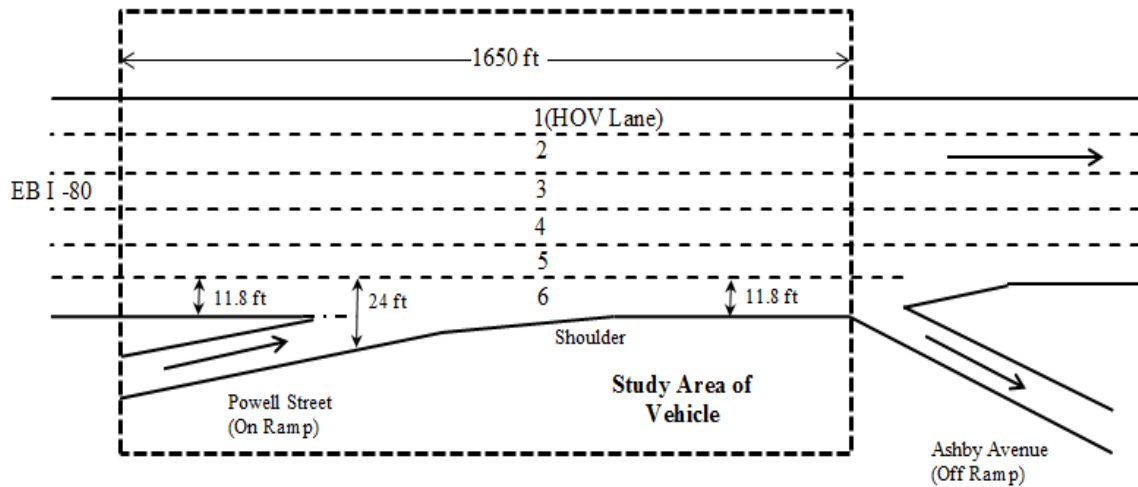


Figure 4.1: Study Area Schematic (NGSIM I-80 Data Analysis Summary report, 2006)

#### 4.2.3 Dataset Overview

The dataset contains detailed trajectory information, observed within the study region over a 45-minute period stretching from 4:00 p.m. to 4:15 p.m. and 5:00 p.m. to 5:30 p.m. The processed dataset presents this information in three parts; the first part encompassing vehicles observed in the first 15-minutes from 4:00 p.m. to 4:15 p.m., the second part for vehicles observed between 5:00 p.m. to 5:15 p.m. and the third part for vehicles observed between 5:15 p.m. to 5:30 p.m. The author used first part of vehicle trajectory for parameter estimation which contains detailed trajectory information of

vehicles observed the study region over 15 minute periods from 4:00 p.m. to 4:15 pm. Complete vehicle trajectories were transcribed at a resolution of 10 frames per second(NGSIM I-80 Data Analysis Summary report, 2006).

A significant proportion of these vehicles (94.6%) were automobiles, as can be seen from the vehicle distribution tables presented NGSIM I-80 Data Analysis Summary report developed by Cambridge Systematics, Inc (NGSIM I-80 Data Analysis Summary report, 2006).

### 4.3 Model Parameter Estimation

Let's consider the Linear (Helly) Model, which is defined by (Bando, M. et al., 1995):

$$a_n(t) = C_1 \Delta v(t-T) + C_2 \Delta x(t-T) - \alpha - \beta v(t-T) - \gamma a_n(t-T)$$

where,  $C_1$ ,  $C_2$ ,  $\alpha$ ,  $\beta$  and  $\gamma$  are the linear model constants to be calibrated. According to this model, the acceleration is a linear function of the speed difference and the difference between headway and desired headway with  $C_1$  and  $C_2$  parameters for the two variables. The desired headway is the function of the velocity and the acceleration of the follow vehicle where  $\alpha$ ,  $\beta$  and  $\gamma$  are parameters for those variables. Therefore, the vector of parameters is,  $\mathbf{q} = (C_1, C_2, \alpha, \beta, \gamma)$ . The authors used synthetic data for the validation of the proposed calibration method utilizing a 500 vehicle trajectory dataset. Then, the Next Generation SIMulation (NGSIM) (4:00PM-4:15PM) database containing observations from 1000 different vehicles with 200 observations for each vehicle on Interstate 80 (I-80) is used for calibration. The NGSIM database represents 45 minutes

(4:00 PM to 4:15 PM, 5:00 PM to 5:30 PM) of data collected during the afternoon peak period on a segment of Interstate 80 in Emeryville (San Francisco), California.

In the following, the authors selected different reaction times,  $T^{(s)}$  for the  $s^{th}$  vehicles with  $s \in \{1, 2, \dots, 1000\}$ , generated randomly from a normal distribution with a mean of 2.2 seconds and a standard deviation of 0.44 seconds, which are given experimentally as good choices (McGehee et al., 2000). The vector can be defined as below,

$$m(q) := \left[ C_1 \Delta v(t - T^{(s)}) + C_2 \Delta x(t - T^{(s)}) - \alpha - \beta v(t - T^{(s)}) - \gamma a_n(t - T^{(s)}) \right]_{n=1, s \in \{1, \dots, M\}}^N$$

where,  $N$  is the number of observations per vehicle for relative velocity, space headway, velocity and acceleration of the follower vehicle and  $M$  is the number of vehicles that are included in the model.

#### 4.3.1 Parameter Estimation Using Synthetic Data

The authors generated synthetic data using a known parameter distribution of a linear model to validate the proposed stochastic calibration method of car-following models. In general, synthetic data can be generated from a known distribution of parameter set and then, the distribution of the parameters should be compared with the distribution of estimated parameter set utilizing synthetic data and proposed calibration method to complete the validation process. In order to generate the distribution of the parameter set, a normal distribution of the parameters was assumed with a given mean (see Figure 4.2) and standard distribution for 500 vehicles. After that, each parameter

was assigned to each of the 500 vehicles where each parameter was constant for all observations for a specific vehicle. Then the observable quantity (e.g. acceleration of the follow vehicle) was calculated using the parameters and synthetic vehicle trajectory data set. Finally, the distribution of calibrated parameters with synthetic data using the proposed Bayesian calibration framework was compared with the normal and uniform prior distribution of parameters. If both of those distribution functions match each other, validation of the calibration method is complete.

Linear model parameters ( $\mathbf{q}$ ) were initially estimated for given observations from 500 different vehicles with given reaction times and normal prior distribution. The authors produced 200,000 random samples of  $\mathbf{q}$ . In Figure 4.3, the convergence of the parameters is shown. The values of the parameters of the linear model over the 200,000 random samples from the Metropolis Hasting algorithm are plotted in Figure 4.3, illustrating the convergence of the proposed Metropolis Hasting algorithm. Generally, the algorithm is considered convergent if the samples look like noisy data around a straight line. A convergence of parameters is also evident in the other cases outlined in this study.

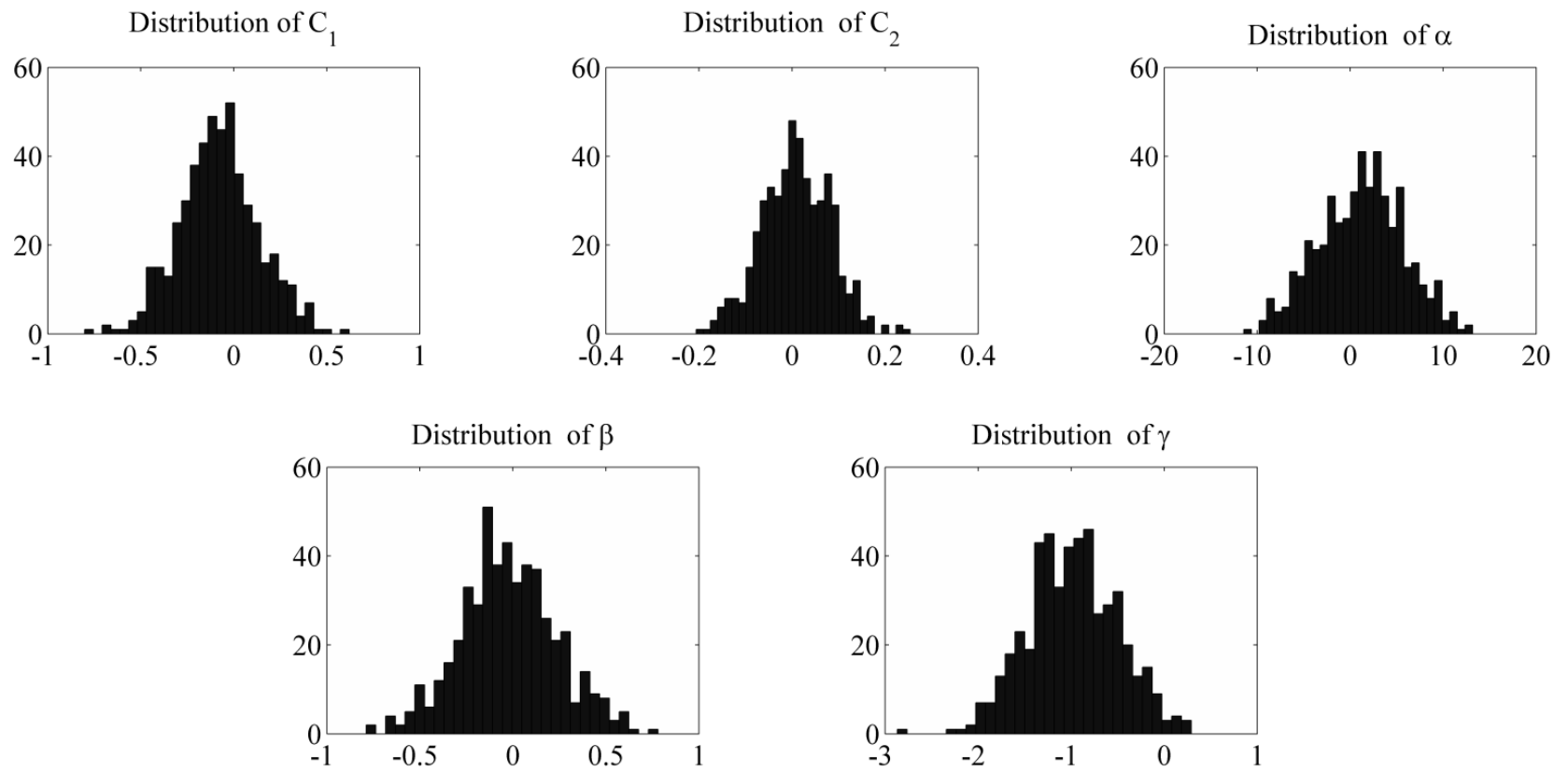


Figure 4.2: Histogram of Parameter for Five Hundred Vehicles using Assumed Distribution of Parameter

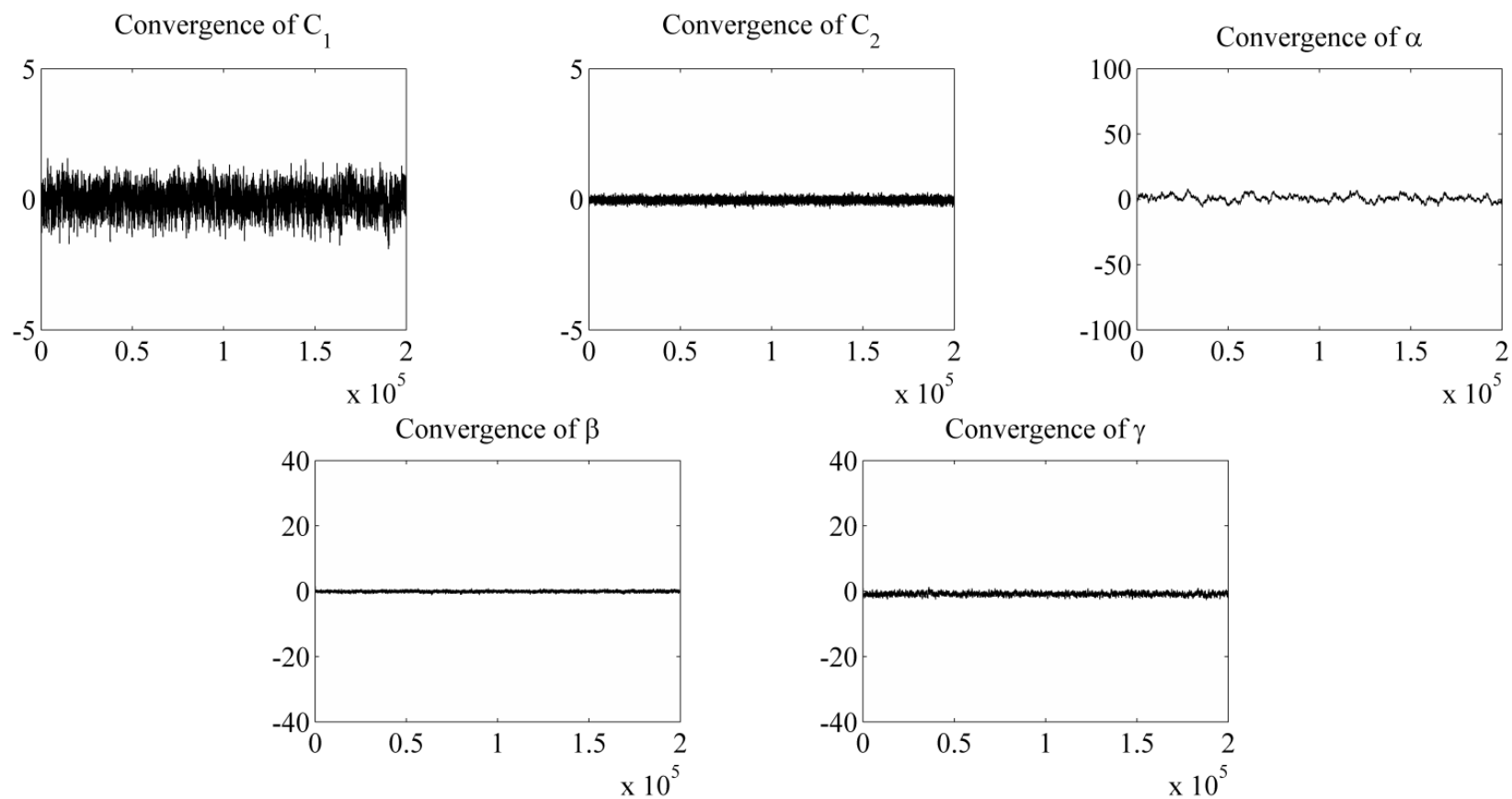


Figure 4.3: Convergence of Parameter with Synthetic Data of Five Hundred Vehicle Trajectories using Normal Prior Distribution



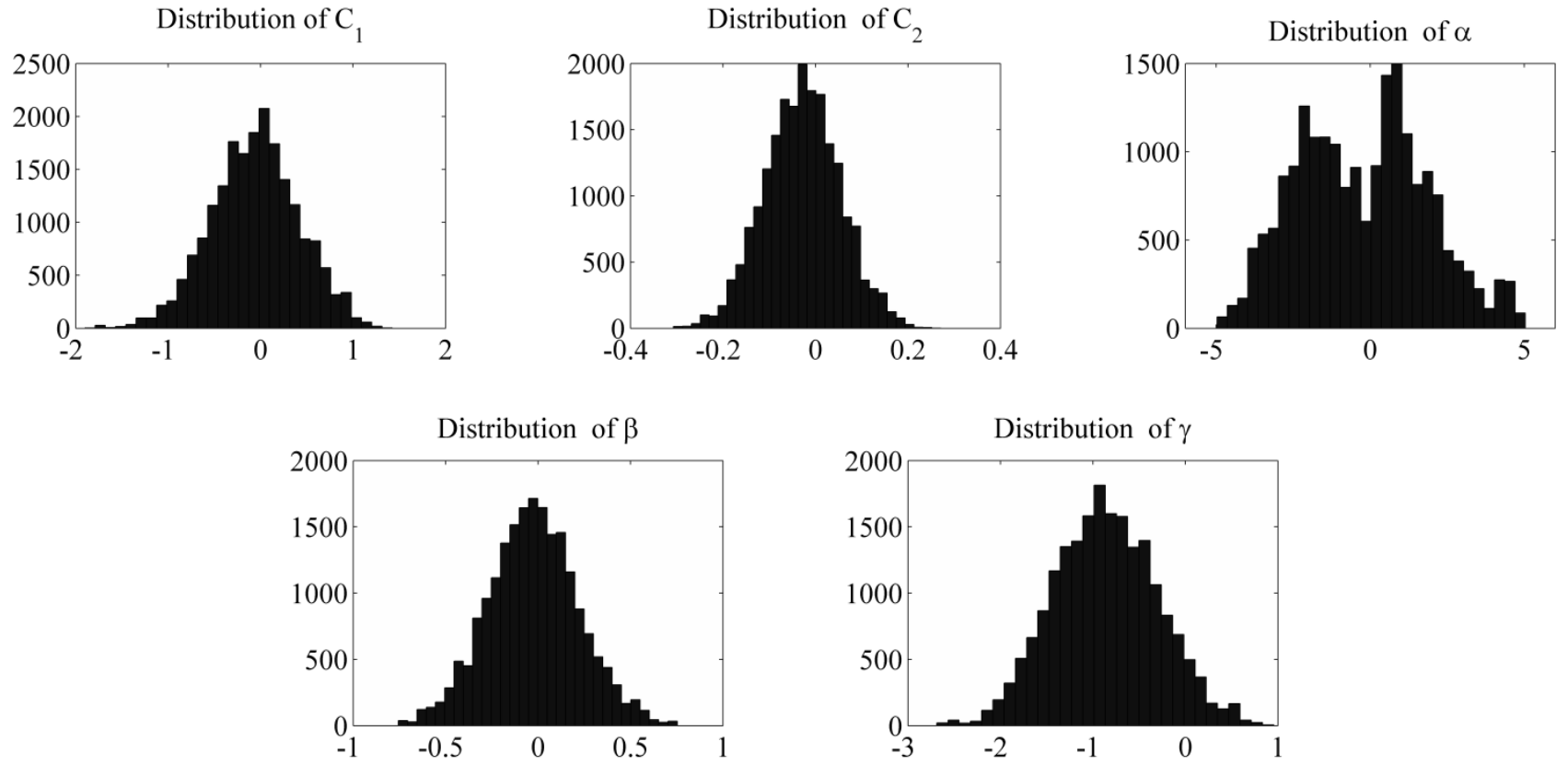


Figure 4.4: Histogram of Parameter with Synthetic Data of Five Hundred Vehicle Trajectories using Normal Prior Distribution

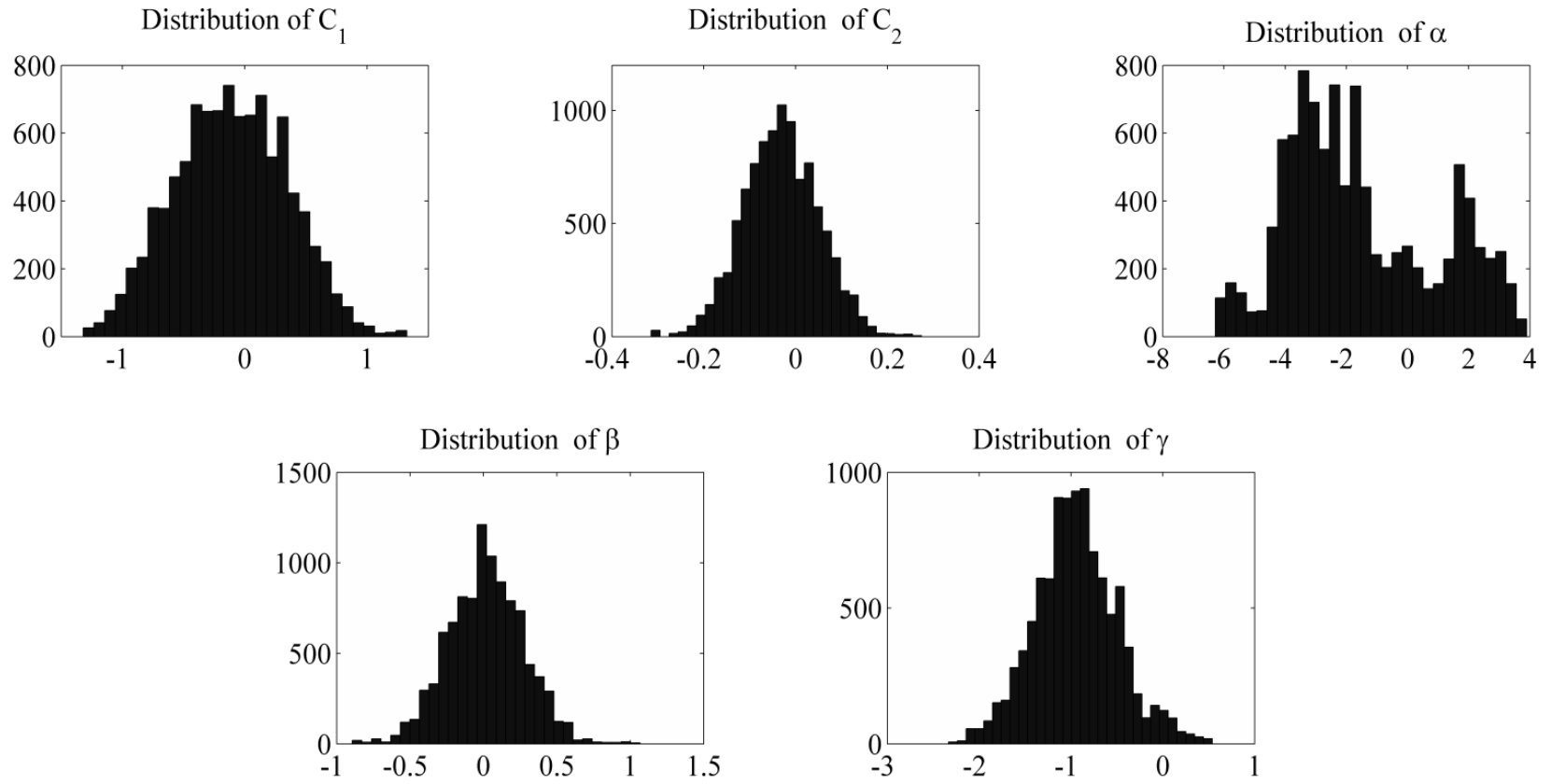


Figure 4.5: Histogram of Parameter with Synthetic Data of Five Hundred Vehicle Trajectories using Uniform Prior Distribution

In Figure 4.4, the distribution function from each of the parameters in  $q$  after a certain burn-in is plotted. The authors plotted the histograms from random samples after a burn-in time of 180,000. This is especially interesting because it visualizes the distribution function from the parameters of the linear (Helly) model given the observations from 500 vehicles. The estimated parameters of the model with normal prior distribution are shown in Table 4-1.

The authors then re-estimated the parameter distributions with prior uniform distribution of the parameter for 500 vehicles to compare with normal prior distribution to observe the effect of both prior distributions on the distribution function of estimated parameters, utilizing the proposed calibration method with the same number of vehicles. For this case, the convergence of parameters was similar as before which is not shown here. Here, 100,000 random samples were created with a burn-in of 90,000 samples. The authors obtained the following distributions (see Figure 4.5) for model parameters. Note that, the distributions of the parameters of the linear model in Figure 4.4 and Figure 4.5 are similar. The mean and standard deviation of the parameters are shown in the Table 4-1.

The distributions appeared more as known distribution functions. Also, the similarity of the distribution function between Figure 4.4 and 4.5 indicates the ability of the proposed calibration method to estimate the parameters from any prior distribution of the parameters of a linear model.

Table 4.1: Validation of Calibration Method

Model Parameters	Mean	Standard Deviation
<b>Generation of synthetic data for 500 vehicles</b>		
$C_1$	-0.0880	0.2045
$C_2$	0.0052	0.0762
$\alpha$	1.1544	4.4685
$\beta$	-0.0283	0.2631
$\gamma$	-1.0060	0.4975
<b>Using normal prior distribution for 500 vehicles</b>		
$C_1$	-0.0589	0.4746
$C_2$	-0.0285	0.0819
$\alpha$	-0.2438	2.1028
$\beta$	-0.0258	0.2447
$\gamma$	-0.8607	0.5630
<b>Using uniform prior distribution for 500 vehicles</b>		
$C_1$	-0.1086	0.4583
$C_2$	-0.0314	0.0820
$\alpha$	-1.5250	2.4184
$\beta$	0.0165	0.2584
$\gamma$	-0.9456	0.4519

#### 4.3.2 Parameter Estimation Using NGSIM Data

Model parameters ( $q$ ) were estimated from ten different vehicles with given reaction times using real-world vehicle trajectory data. The author produced 15,000,000 random samples of  $q$ . In Figure 4.6, the convergence of the parameters is shown. The values of the parameters of the linear model over the 15,000,000 random samples from the proposed Metropolis Hasting algorithm are plotted in Figure 4.6.

In Figure 4.7, the distribution function from each of the parameters in  $q$  after a certain burn-in is plotted. In Figure 4.7, the authors plotted the histograms from random samples after a burn-in time of 1350,000. This is especially interesting because it visualizes the distribution function from the parameters of the linear (Helly) model given the observations from 10 vehicles. In this case, the authors obtained the approximation of the Bayes estimate:

$$E(q|d_{10 \text{ vehicles}}) \approx (-0.1029, 0.0019, -0.9118, 0.0802, -0.9077)$$

The authors then re-estimated the parameter distributions with observations from 500 vehicles, to compare with the Bayes estimate from 500 vehicles to observe the effect of estimating the parameter distribution with more observations. For this case, convergence of parameters was similar as before. Here, 1,500,000 random samples were generated with a burn-in of 1,350,000 samples. The authors obtained the following distributions (see Figure 4.8) for model parameters. Compared to Figure 4.7, the distributions of the parameters of the linear model are changing if the observations of 500 instead of only 10 vehicles are considered. The following approximation from the Bayes estimate expresses that thought:

$$E(q|d_{500 \text{ vehicles}}) \approx (-0.0880, 0.0052, 1.1544, -0.0283, -1.0060)$$

In Figure 4.5, the distributions of the linear model (Helly's model) parameters considering all observations from 1,000 vehicles are plotted. It is worth noting that the distribution of  $C_1$ ,  $C_2$ ,  $\beta$  and  $\gamma$  are similar as shown in Figures 4.7, 4.8, and 4.9 and

seem to have normal distributions. The authors tested the normality of these four parameters and they passed the normality test with a small number of outliers. The distribution of parameter  $\alpha$  didn't follow the normal distribution as the other four parameters. The difference between distributions of parameter  $\alpha$  in Figures 4.8 and 4.9 is much smaller than the difference between Figures 4.7 and 4.8. This difference concludes that the more observations from vehicles included, the closer the approximation comes to the real distribution of this linear model parameter. The authors obtained the following Bayes estimate given the observations from all 1000 vehicles:

$$E(\mathbf{q} | d_{1000 \text{ vehicles}}) \approx (-0.0933, -0.0051, -2.0020, 0.0651, -1.0279)$$

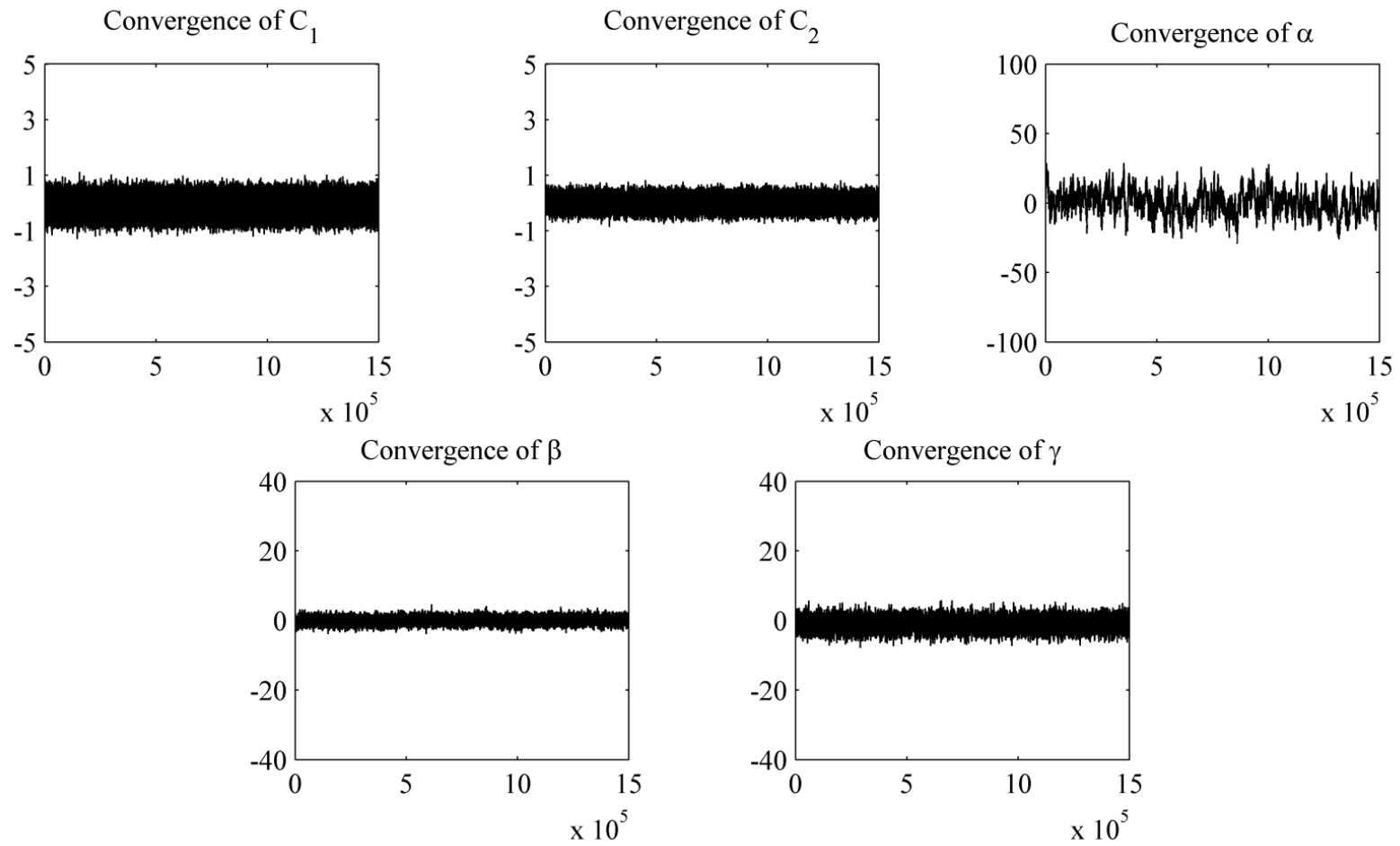


Figure 4.6: Convergence of Parameter with Ten (10) Vehicle Trajectories

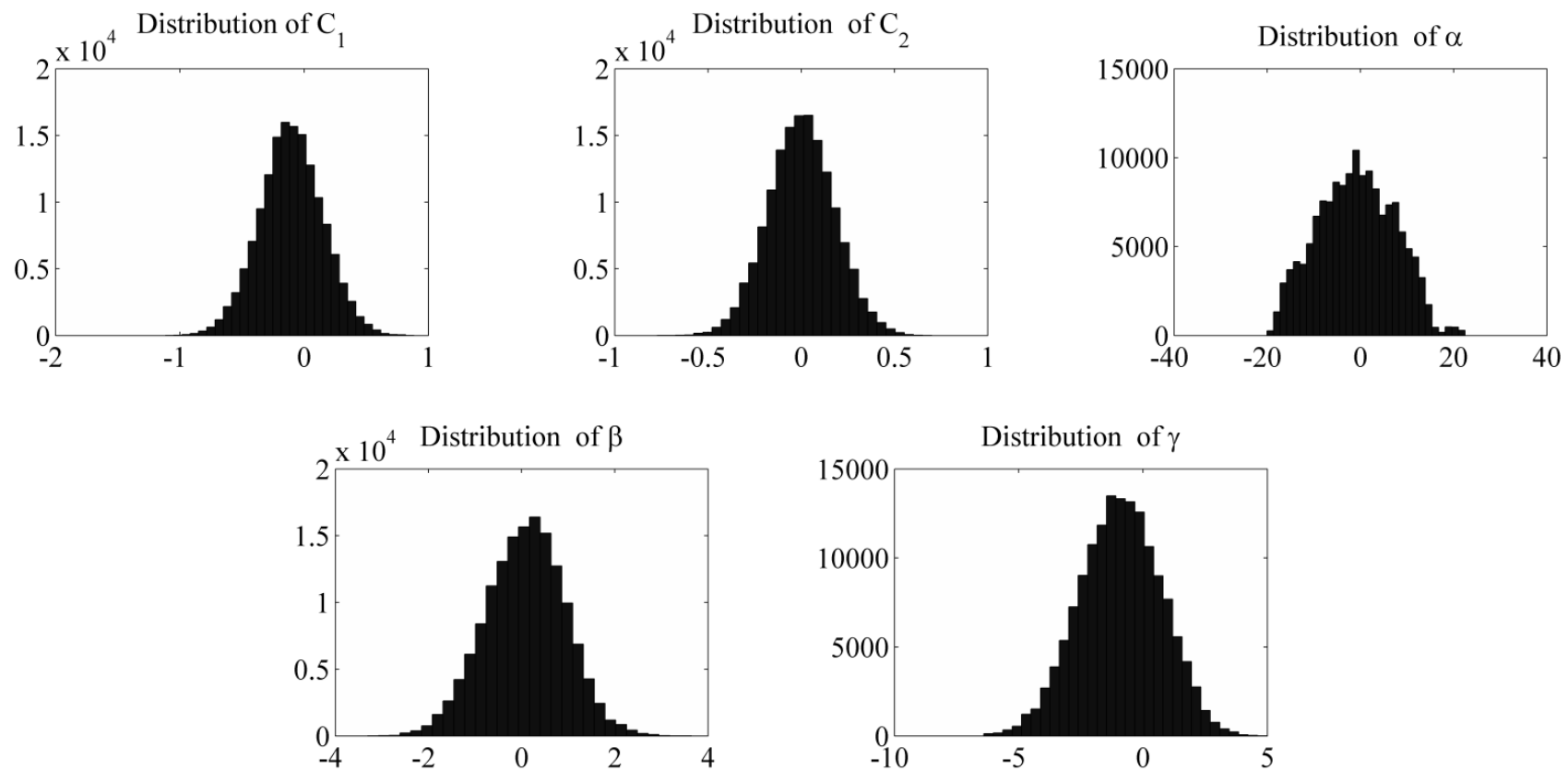


Figure 4.7: Histogram of Parameter with Ten (10) Vehicle Trajectories



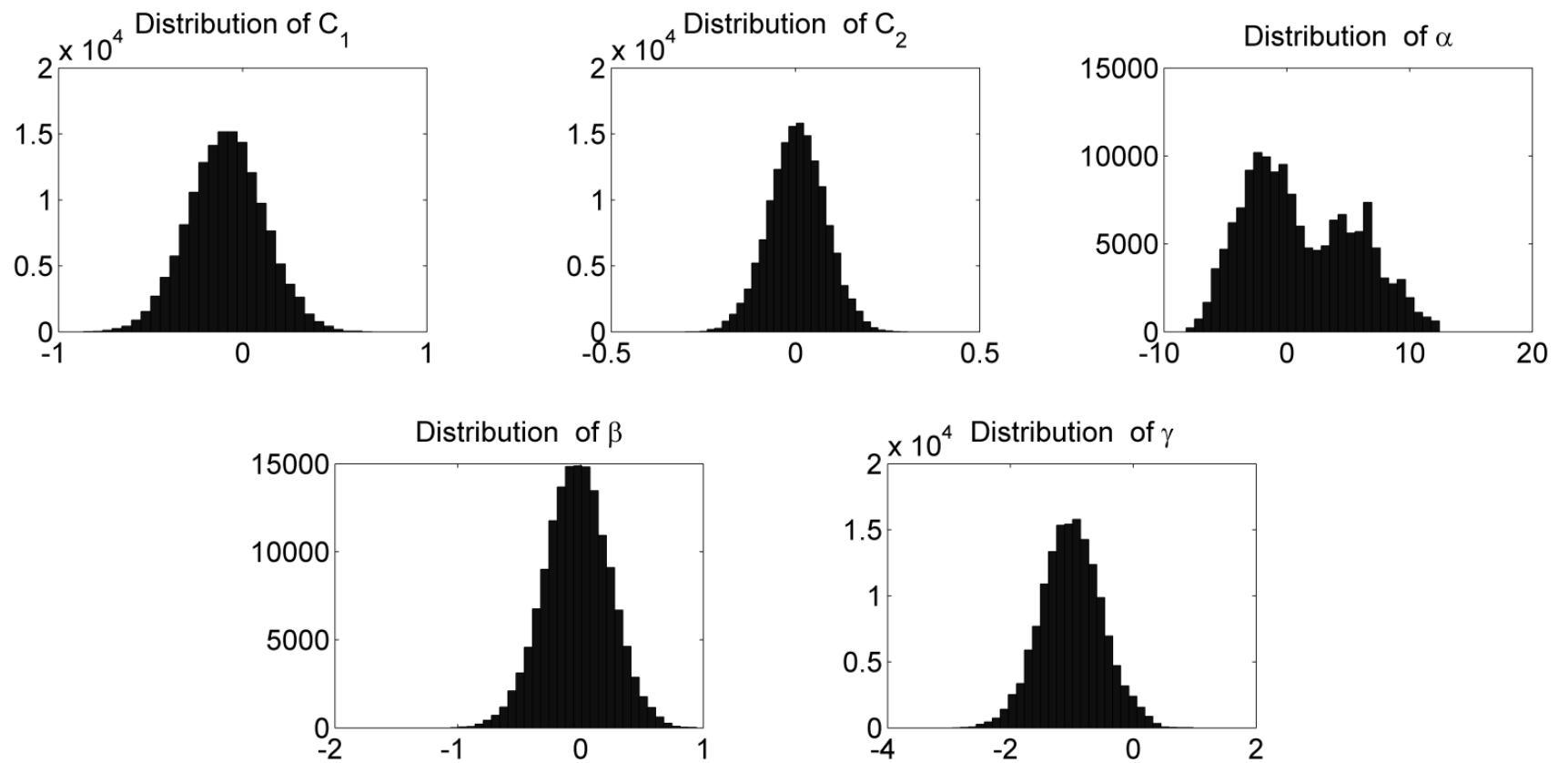


Figure 4.8: Histogram of Parameter with Five Hundred (500) Vehicle Trajectories

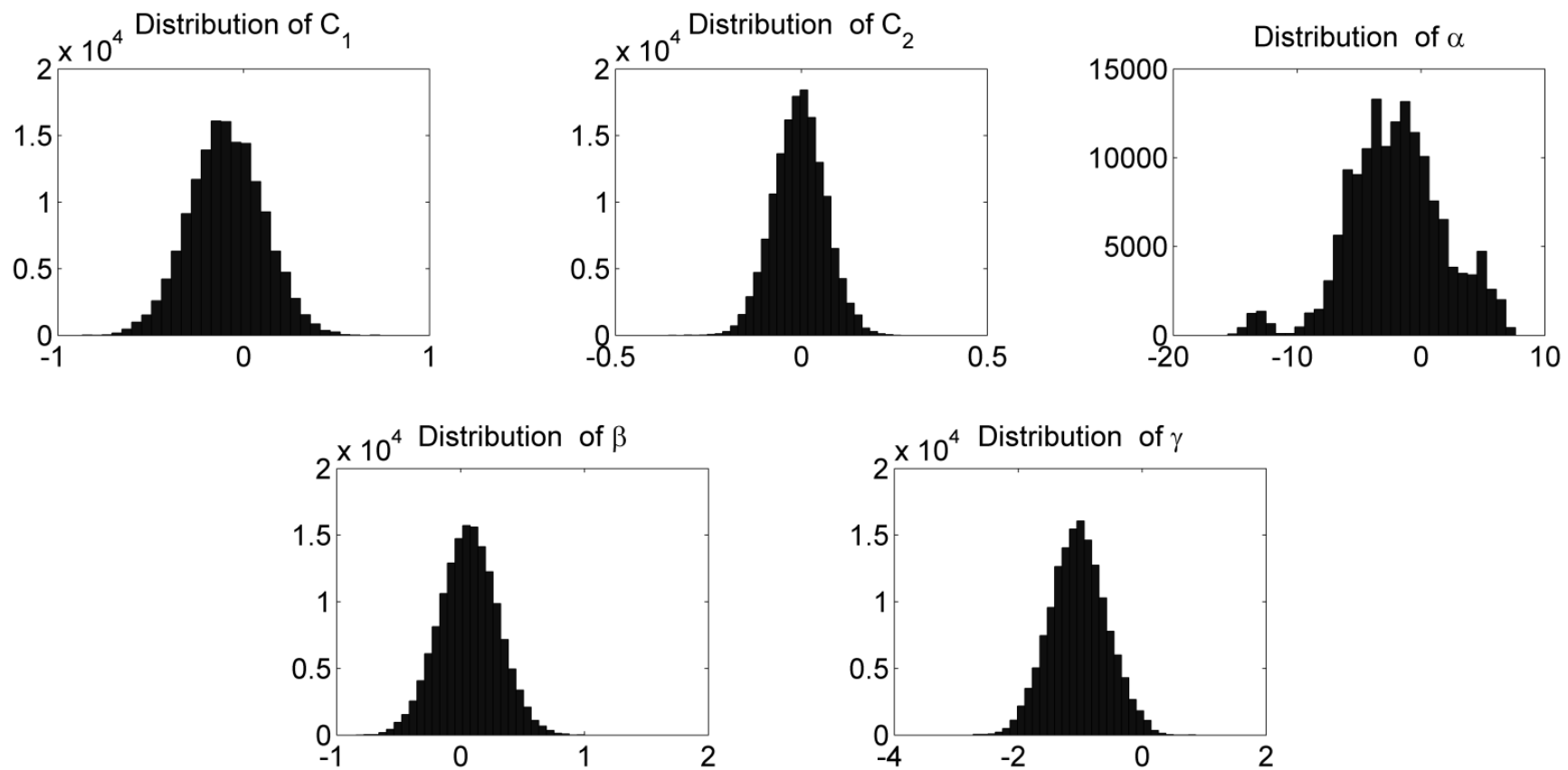


Figure 4.9: Histogram of Parameter with One Thousand (1000) Vehicle Trajectories

The Bayesian framework was deemed computationally efficient for 10 vehicles after having been simulated 15,000,000 times. The simulation time was 2.56 hours to generate each 500,000 random sample. The simulation running time largely depends on the efficiency of the MATLAB coding for the Metropolis-Hasting algorithm for a given dataset, number of observations taken for each vehicle, configuration of the computer, and most importantly, the prior distribution of the parameters. In cases of 500 and 1000 vehicles, the simulation time for generating 100,000 samples was 7.54 hours and 15.40 hours, respectively. The authors have generated 15,000,000 samples for 10, 500 and 1000 vehicles to observe convergence of the parameters. Future research should include possible modifications in the Bayesian framework presented in this study to decrease the number of iterations required in order to improve the calibration efficiency.

#### **4.4 Calibration Results**

A summary of means and standard deviations of the parameters for each vehicle set of the linear model is shown in Table 4-2. Note that the trend of the model parameter values is not similar for each vehicle set. Although, the distribution of parameters,  $C_1$ ,  $C_2$ ,  $\beta$  and  $\gamma$  follows the normal distribution, parameter  $\alpha$  does not look like a normal distribution. Since, parameter,  $\alpha$  is not directly related to any car-following model variables, it doesn't follow the normal distribution. The mean and standard deviation of all the parameters are calculated for the validation purpose. Prior distribution (initial guess of mean and standard deviation of model parameters) plays an important role to obtain better convergences and normal distributions of the parameters after a certain burn-in time.

Table 4.2: Model Parameters Summary

Model Parameters	Mean	Standard Deviation
<b>For 10 vehicles</b>		
$C_1$	-0.1029	0.2530
$C_2$	0.0019	0.1772
$\alpha$	-0.9118	8.1835
$\beta$	0.0802	0.8459
$\gamma$	-0.9077	1.6423
<b>For 500 vehicles</b>		
$C_1$	-0.0880	0.2045
$C_2$	0.0052	0.0762
$\alpha$	1.1544	4.4685
$\beta$	-0.0283	0.2631
$\gamma$	-1.0060	0.4975
<b>For 1000 vehicles</b>		
$C_1$	-0.0933	0.1975
$C_2$	-0.0051	0.0689
$\alpha$	-2.0020	3.9201
$\beta$	0.0651	0.2349
$\gamma$	-1.0279	0.4574

#### 4.5 Evaluation

In order to consider whether this is a good (or meaningful) method, the author calculated the average mean square error for one vehicle (taking 200 observations per vehicle), with each of the three Bayes estimates and the author also compared it to the "optimal" parameters found through the deterministic method, which suggests that the Bayesian calibration method provides smaller error than the deterministic calibration

method. The average mean square error per vehicle is shown in Table 4.3 for the three Bayes estimates for each vehicle set, and with the parameters of the deterministic method. In Table 4.3 it is clear that the calibration of the given observations from all 1000 vehicles gives, at an average, the smallest mean square error per vehicle. The average Mean Square Error (MSE) per vehicle decreased with increasing number of vehicles used to estimate the model parameters with the Bayes calibration method. Thus, the performance of the calibration method is dependent upon the sample size.

Table 4.3: Average Mean Square Error (MSE) per Vehicle for Calibration

Estimation Approach	Average MSE per vehicle
$E(\hat{q})$ (Deterministic approach)	5186.09
$E(q d_{10 \text{ vehicles}})$	286.38
$E(q d_{500 \text{ vehicles}})$	105.27
$E(q d_{1000 \text{ vehicles}})$	92.01

The Bayesian approach provides better results than deterministic optimization algorithms ( $\hat{q}$ ). It seems logical that the more observations from vehicles that are given, the better our calibration. Furthermore, this method is superior in that it is possible to estimate the distribution of the parameters rather than just the mean as in the deterministic approaches. Figure 4.10 represents acceleration/deceleration profiles of estimated data and observed data for a randomly selected vehicle. With an increasing number of vehicles, the average mean square error per vehicle is decreases, and the acceleration/deceleration profile is closer to the observed profile of the given field data.

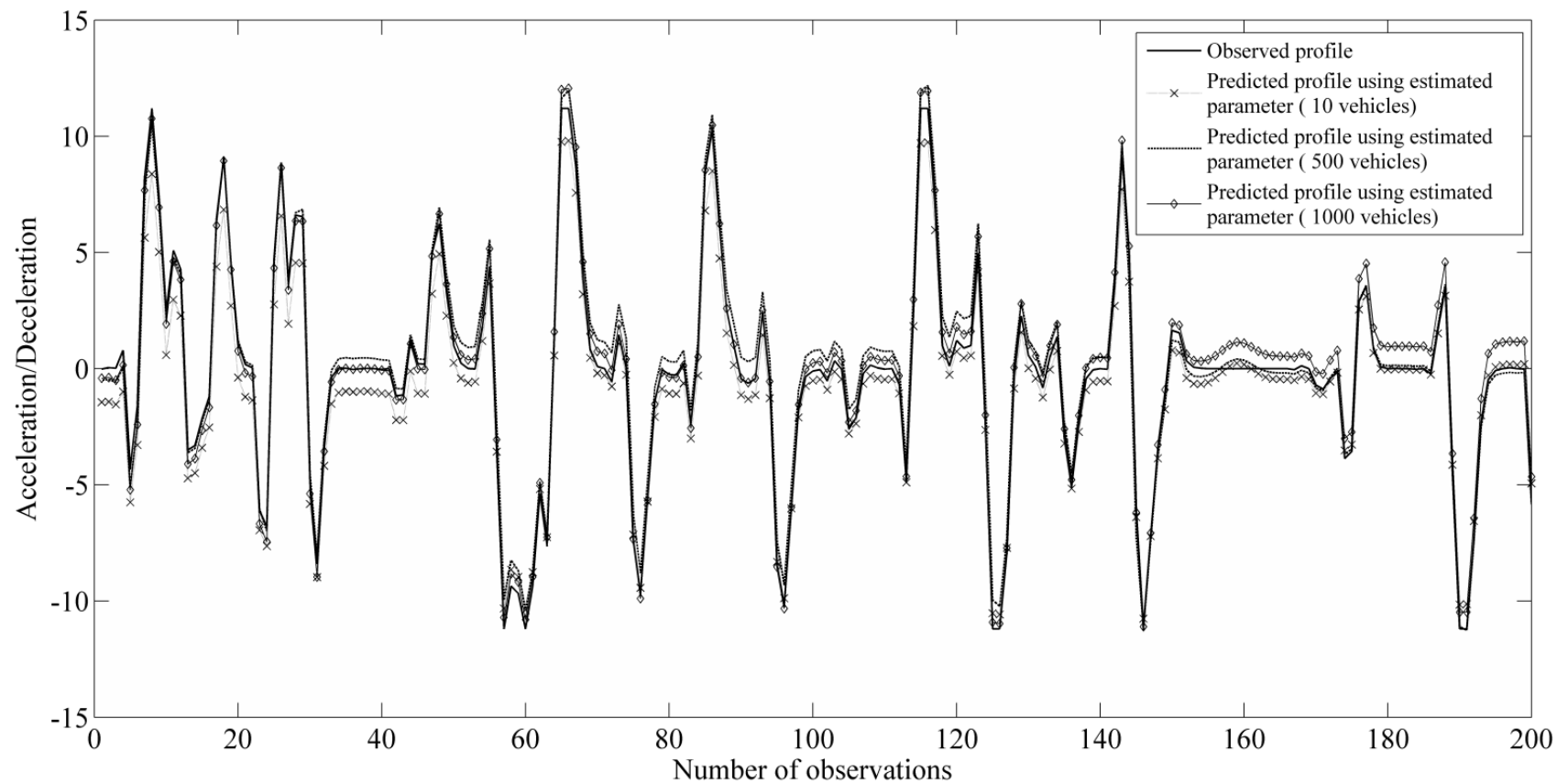


Figure 4.10 Comparison of Acceleration/Deceleration Profile Among Estimated Data and Observed Data for Calibration

## 4.6 Validation Results

Using a comparative model validation, the average mean square error per vehicle for three different Bayes estimates was selected to measure the performance of the Bayesian calibration method. In Table 4.4, average mean square error per vehicle was calculated with data from Interstate 80 (I-80) that were collected by the Next Generation SIMulation (NGSIM) (time period 5:00PM-5:15PM). One thousand vehicles are randomly selected from this database to calculate the average mean square error. The average mean square error per vehicle decreased with increasing a number of vehicles used to estimate the model parameters using the Bayes calibration method. Thus, the performance of the calibration method is dependent upon the sample size.

Table 4.4: Average Mean Square Error (MSE) per Vehicle for Validation

Estimation Approach	Average MSE per vehicle
$E(q d_{10 \text{ vehicles}})$	502.59
$E(q d_{500 \text{ vehicles}})$	86.66
$E(q d_{1000 \text{ vehicles}})$	75.81

To further investigate the performance, predicted acceleration/deceleration profile with three different Bayes estimates, and observed data for a randomly selected vehicle for 200 observations is compared (See Figure 4.11). With an increasing the number of vehicles, the average mean square error per vehicle is decreases and the acceleration/deceleration profile is closer to the observed profile of the given field data. From Figure 4.11, one can recognize relatively feasible behavior that the model predicts

regarding drivers' acceleration and deceleration behavior and consequently map to the field data.



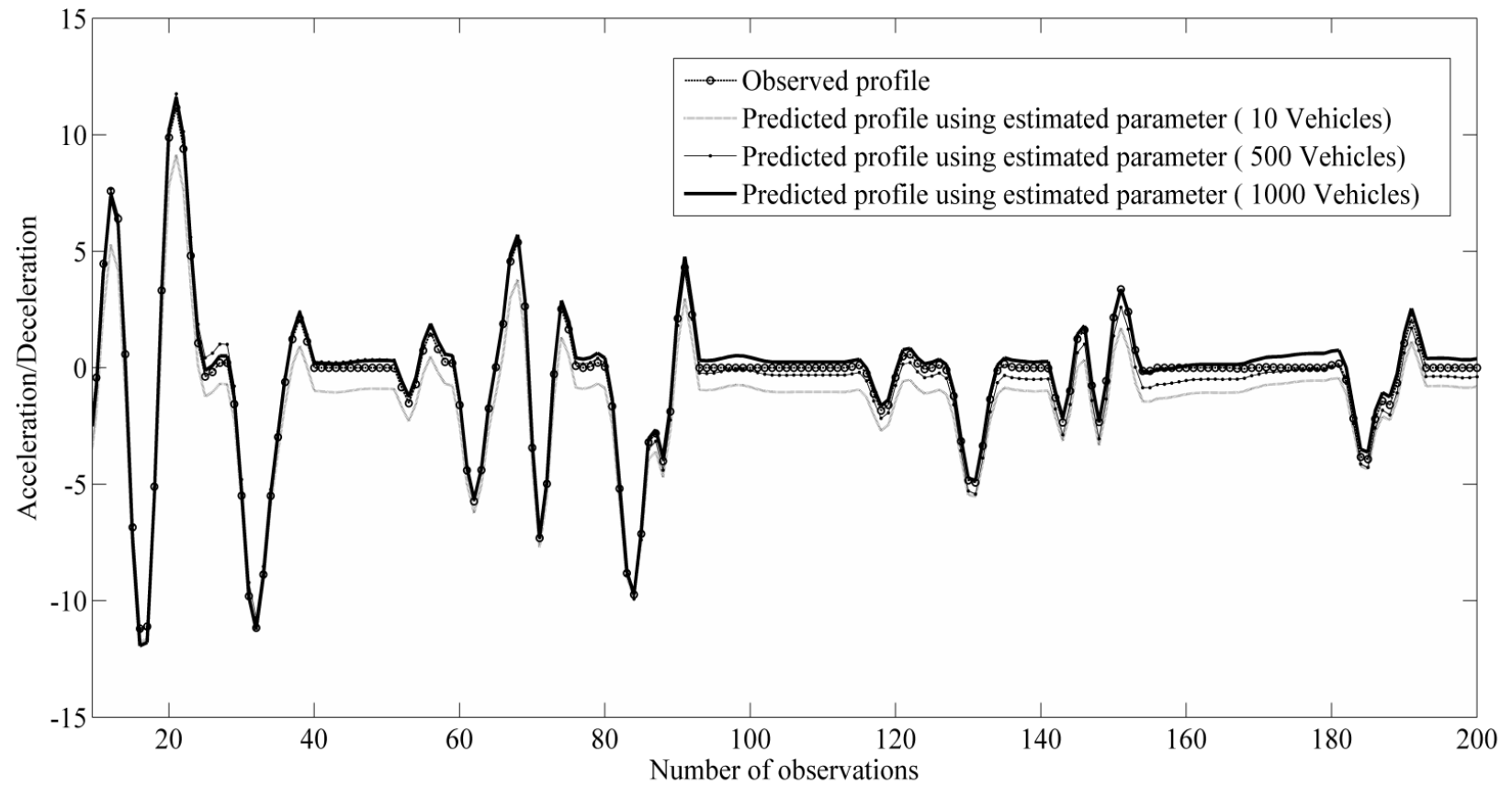


Figure 4.11: Comparison of Acceleration/Deceleration among Predicted Data and Observed Data for Validation

#### **4.7 Contribution of the Research**

This research focused on the development of a method to apply a stochastic calibration approach to car-following models. In this study, a stochastic calibration method was developed utilizing a Bayesian framework, which is based on Markov Chain Monte Carlo (MCMC) simulation to estimate the parameters of a car-following model. This stochastic method will facilitate the calibration of car-following model more realistically than the deterministic methods. This calibration method was applied to estimate the parameters in a linear car-following model utilizing real world data from the NGSIM database. This study demonstrated that with increasing sample size, calibrated model would produce smaller errors. The calibration method presented in this thesis provided better results than the deterministic optimization algorithm considered in this study. This thesis will support the real world applications of car-following models in representing driver behaviors; thus supporting more realistic simulations and evaluations of roadway traffic.

## CHAPTER FIVE

### CONCLUSIONS AND RECOMMENDATIONS

#### 5.1 Conclusions

The primary goal of this research was to develop a process for applying a stochastic calibration method with appropriate regularization to estimate the distribution of parameters for car-following models. The calibration method was based on the Markov Chain Monte Carlo (MCMC) simulation utilizing Bayesian estimation theory that has been recently investigated for inverse problems. This research proved the efficacy of the proposed approach using a synthetic dataset to estimate the parameters of a linear model with both normal and uniform prior distribution of the parameters.

The calibration method was then applied to a relatively simple car-following model (Linear or Helly model) to provide a comparison with a deterministic approach. The analysis revealed that the calibration of the parameters of the linear model, given the distribution from all 1000 vehicles, on average yielded the smallest mean square error per vehicle. On the other hand, the deterministic calibration approach provided a higher mean square error per vehicle than the Bayesian framework. Thus, the Bayesian approach provided better results in terms of the cost function than the deterministic optimization algorithm. It was also determined that the stochastic approach facilitated the calibration of car-following models more realistically than the deterministic methods, as the deterministic algorithm can easily get stuck at a local minimum.

The Bayesian framework was deemed computationally efficient for 10 vehicles after having been simulated 15,000,000 times. The simulation time was 2.56 hours to generate each 500,000 random sample. The simulation running time largely depends on the efficiency of the MATLAB coding for the Metropolis-Hasting algorithm for a given dataset, number of observations taken for each vehicle, configuration of the computer, and most importantly, the prior distribution of the parameters. In cases of 500 and 1000 vehicles, the simulation time for generating 100,000 samples was 7.54 hours and 15.40 hours, respectively. The authors have generated 15,000,000 samples for 10, 500 and 1000 vehicles to observe convergence of the parameters. Future research should include possible modifications in the Bayesian framework presented in this study to decrease the number of iterations required in order to improve the calibration efficiency.

Of particular interest were the trends of the three Bayes estimates of the linear model parameters that were very close from each other. Additional running time may be necessary to get a better approximation of the parameters. Since the calibration process of the Bayesian framework depends on the vehicle trajectory dataset and simulation to generate random samples to converge, the limitations of this research include the long computational time of the calibration process in the simulation and a large vehicle trajectory dataset requirement to provide the most reliable results. In summary, the stochastic calibration approach has been rigorously validated in this research with synthetic data. As heavy computational burden is one of the major limitations, a linear model was used to overcome this issue.

## 5.2 Recommendations

This section is divided into two sections. The first section presents recommendations regarding the application of the framework to other models and the second section presents recommendations regarding follow-up research.

### 5.2.1 Applications of the Framework

- ❖ Any application of the framework should strive to include a large number of vehicle trajectories as the analysis conducted for this thesis suggests that a larger number of vehicle trajectories resulted in smaller errors in the calibrated model.
- ❖ The calibration framework presented in this thesis will be more precise as more accurate vehicle trajectories are generated through the real-time tracking of vehicles.

### 5.2.2 Future Research

- ❖ Future research should include possible modifications in the Bayesian framework presented in this thesis, to decrease the number of iterations required, in order to improve the calibration efficiency.
- ❖ Future research should investigate the calibration efficacy under different driving conditions, such as traffic incidents on the roadway.

## REFERENCES

- Ahmed, K. I., Ben-Akiva, M., Koutsopoulos, H. N., and Mishalani, R. G., "Models of Freeway Lane-changing and Gap Acceptance Behavior," Proceedings of the 13th International Symposium on the Theory of Traffic Flow and Transportation, 1996.
- Ahmed, K. I., "Modeling Drivers' Acceleration and Lane-changing Behavior," MS Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Feb. 1999.
- Aumann, R. J., "Lectures on Game Theory," Westview Press Inc., Boulder, Colorado, 1989.
- Bando, M., Hasebe, K., Nakayama, A., Shibata, A., and Sugiyama, Y., "Dynamical Model of Traffic Congestion and Numerical Simulation". Physical Review E, Vol. 51, No. 2, pp. 1035-1042, 1995.
- Brackstone, M., and McDonald, M. "Car-Following: A Historical Review". Transportation Research Part F: Traffic Psychology and Behavior, Vol. 2, No. 4, pp. 181-196, 1999.
- Brackstone, M., McDonald, M. , and Wu, J., "Lane-changing on the motor- 1245way: Factors affecting its occurrence, and their implications," in Proc. 9th 1246 Int. Conf. Road Transp. Inf. Control, London, U.K, pp. 160–164., 1998.
- Chakroborty, P., and Kikuchi, S., "Evaluation of the General Motors Based Car-Following Models and a Proposed Fuzzy Inference Model". Transportation Research Part C: Emerging Technologies, Vol. 7, No. 4, pp. 209-235, 1999.
- Chandler, R. E., Herman, R., and Montroll E. W., "Traffic Dynamics: Studies in Car Following". Operations Research, pp. 165-184, 1958.
- Choudhury, C. F., Ben-Akiva, M., E., Toledo, T., Lee, G.,and Rao, A., "Modeling Cooperative Lane-changing and Forced Merging Behavior," Proceeding of the 86th Transportation Research Board Annual Meeting. Washington, DC. CD-ROM, 2007.
- Das, S., and Bowles, B. A., "Simulations of Highway Chaos Using Fuzzy Logic," Proc., 18th International Conference of the North American Fuzzy Information Processing Society, pp. 130–133, Jul. 1999.

- Das, S., and Bowles, B. A., Houghland, C. R., Hunn, S. J., and Zhang, Y., "A Fuzzy Logic Model of Freeway Driver Behavior," Proc., International ICSC Congress on Computational Intelligence Methods and Applications, New York, 1999.
- Domencich T. A., and McFadden, D., "Urban Travel Demand: A Behavioral Analysis," North-Holland, Amsterdam. Japan Society of Traffic Engineers, 1989. Research Report on the Design of Merging Sections, No. 1, No. 2, 1988, No. 3. (In Japanese), 1987.
- Doshi, A. and Trivedi, M., "A Comparative Exploration of Eye Gaze and Head Motion Cues for Lane Change Intent Prediction". IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands: 49-54, 2008.
- Dumbuya, A., Booth, A., Reed, N., Kirkham, A., Philpott, T., Zhao, J., and Wood, R., "Complexity of Traffic Interactions: Improving Behavioural Intelligence in Driving Simulation Scenarios," Complex Systems and Self-organization Modelling, 201–209, 2009.
- Eidehall, A., Pohl, J., Gustafsson, F., and Ekmark, J., "Toward Autonomous Collision Avoidance by Steering". *IEEE Transactions on Intelligent Transportation Systems*, 8(1): 84-94, 2007.
- Gazis, D. C., Herman, R., and Potts R. B., "Car-Following Theory of Steady-State Traffic Flow". *Operations Research*, pp. 499-505, 1959.
- Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B., "Bayesian Data Analysis," CRC press, 2004.
- Gipps, P. G. "A Behavioral Car-Following Model for Computer Simulation" *Transportation Research Part B: Methodological*, Vol. 15, No. 2, pp. 105-111, 1981.
- Gipps, P. G., "A Model for the Structure of Lane-changing Decisions," *Transportation Research B* 20(5): 403-414, Oct. 1986.
- Gong, H., Liu, H., and Wang, B., "An Asymmetric Full Velocity Difference Car-Following Model". *Physica A: Statistical Mechanics and its Applications*, Vol. 387, No. 11, pp. 2595-2602, 2008.
- Halati, A., Lieu, H., and Walker, S., "CORSIM – Corridor traffic simulation model," *Proceedings of the Traffic Congestion and Traffic Safety in the 21st Century Conference*, pp. 570-576, 1997.

- Helbing, D., and Tilch, B., "Generalized Force Model of Traffic Dynamics". *Physical Review E*, Vol. 58, No. 1, pp. 133-138, 1998.
- Helly, W., "Simulation of Bottlenecks in Single Lane Traffic Flow". *Theory of Traffic Flow*, pp. 207-238. 2, 1961.
- Herman, R., and Potts, R. B., "Single-Lane Traffic Theory and Experiment". In *Symposium on Theory of Traffic Flow*, pp. 120-146, 1961.
- Hidas, P. and Behabahanizadeh, K., "Urban transport network simulation: an object-oriented approach," In: Wyatt, R., Hossein, H. (Eds.), *Proceedings of the 4th International Conference on Computers in Urban Planning and Urban Management*, vol. 2, Melbourne, Australia, pp. 229–243, 1995.
- Hidas, P. and Behabahanizadeh, K., "SITRAS: A simulation model for ITS applications," in *Proc. 5th World Congr. Intell. Transp. Syst.*, Seoul, Korea, Oct. 1998.
- Hidas, P. "Modelling lane changing and merging in microscopic traffic simulation," *Transp. Res. Part C*, vol. 10, no. 5/6, pp. 351–371, Oct.–Dec. 2002.
- Hidas, P., "Modelling Vehicle Interactions in Microscopic Simulation of Merging and Weaving," *Transportation Research C* 13(1): 37–62, Feb. 2005.
- Holm P. and Tomich, D., "Traffic Analysis Toolbox: Volume IV: Guidelines for Applying CORSIM Microsimulation Modeling Software". Washington, DC, USA: U.S. Dept. Transp. Fed. Highway Admin., Jan. 2007. [Online]. Available: [http://ops.fhwa.dot.gov/trafficanalysistools/tat\\_vol4/app\\_i.htm](http://ops.fhwa.dot.gov/trafficanalysistools/tat_vol4/app_i.htm)
- Hoogendoorn, S. P., and Hoogendoorn, R., "Generic Calibration Framework for Joint Estimation of Car-Following Models by using Microscopic Data". *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2188, No. -1, pp. 37-45. 25, 2010.
- Hunt, J. G., and Lyons, G. D., "Modeling Dual Carriageway Lane-changing Using Neural Networks," *Transportation Research Part C*, Vol. 2, No. 4, pp. 231–245, 1994.
- IntelliDriveSM. USDOT. <http://www.its.dot.gov/intellidrive/>, Accessed 1/13, 2011.
- Interstate 80 Freeway Dataset. Federal Highway Administration. Accessed 03/18/2012. <http://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm>, December 2006.
- Jiang, R., Wu, Q., and Zhu, Z., "Full Velocity Difference Model for a Car-Following Theory". *Physical Review E*, Vol. 64, No. 1, pp. 17101, 2001.



- Kaipio, J., and Somersalo, E., "Statistical and Computational Inverse Problems". Springer Verlag, 2005.
- Kesting, A., and Treiber, M., "Calibrating Car-Following Models by using Trajectory Data: Methodological Study". Transportation Research Record: Journal of the Transportation Research Board, Vol. 2088, No. -1, pp. 148-156. 21, 2008.
- Kiefer, R. J. and Hankey, J. M., "Lane Change Behavior with a Side Blind Zone Alert System". Accident Analysis and Prevention, 40(2): 683-690, 2008.
- Kikuchi, S., and Chakroborty, P., "Car-Following Model Based on Fuzzy Inference System". 1992.
- Kita, H., "A Merging-Give-Way Interaction Model of Cars in a Merging Section: A Game Theoretic Analysis," Transportation Research Part A, Vol. 33, pp. 305–312, Apr. 1999.
- Kita, H., "Effects of merging lane length on the merging behavior at expressway on-ramps," In: Daganzo, C. (Ed.), Transportation and Traffic Theory. Elsevier, Amsterdam pp. 37-51, 1993.
- Kometani, E., and Sasaki, T., "Dynamic Behavior of Traffic with a Nonlinear Spacing-Speed Relationship". Theory of Traffic Flow, pp. 105-119, 1959.
- Li-sheng, J., Wen-ping, F., Ying-nan, Z., Shuang-bin, Y. and Hai-jing, H., "Research on Safety Lane Change Model of Driver Assistant Aystem on Highway". IEEE Intelligent Vehicles Symposium, Shaanxi, China, 2009.
- Lygeros, J., Godbole, D. N., Sastry, S., "Verified Hybrid Controllers for Automated Vehicles". IEEE Transactions on Automatic Control 43(4): 522-539, 1998.
- Ma, X., "Toward an Integrated Car-following and Lane-changing Model Based on Neural-fuzzy Approach," Helsinki summer workshop, 2004.
- Maerivoet, S., and Moor, B. D., "Cellular Automata Models of Road Traffic". Physics Reports 419(1): 1-64, 2005.
- McDonald, M., Wu, J., and Brackstone, M., "Development of a Fuzzy Logic Based Microscopic Motorway Simulation Mode," Proceedings of the IEEE Conference on Intelligent Transportation Systems, Boston, U.S.A, Nov. 1997.
- McGehee, D. H., Mazzae, E. N., and Baldwin, G. H. S., "Driver Reaction Time in Crash Avoidance Research: Validation of a Driving Simulator Study on a Test Track,"

- In Proceedings of Human Factors and Economics Society Annual Meeting, SAGE Publications, pp. 3-320-3-323, 2000.
- Mendel, J. M., “Fuzzy Logic Systems for Engineering: A Tutorial,” Proceedings of the IEEE, Vol. 83, No. 3, pp. 345–377, 1995.
- Moridpour, S., Sarvi, S., Rose G., and Mazlami, E., “Lane-Changing Decision Models for Heavy Vehicles Drivers,” Journal of Intelligent Transportation Systems: Technology, Planning, and Operations. Volume 16, Issue 1, pages 24-35, Feb. 2012.
- Nagel, K., Wolf, D. E., Wagner, P., and Simon, P., “Two-lane traffic rules for cellular automata: a systematic approach,” Physical Review E, 58, pp. 1425–1437, Aug. 1998b.
- NGSIM I-80 Data Analysis, Summary report. Cambridge Systematics, Inc. 2006.
- Panwai, S., and Dia, H., “Comparative Evaluation of Microscopic Car-Following Behavior”. IEEE Transactions on Intelligent Transportation Systems, Vol. 6, No. 3, pp. 314-325, 2005.
- Pei, Y., and Xu, H., “The Control Mechanism of Lane-changing in Jam Condition,” Proc., 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 8655–8658, 2006.
- Peng, G., and Sun, D., “A Dynamical Model of Car-Following with the Consideration of the Multiple Information of Preceding Cars”. Physics Letters A, Vol. 374, No. 15-16, pp. 1694-1698, 2010.
- Reijmers, J., “Traffic Guidance Systems”. <http://www.pn.ewi.tudelft.nl/education/et4-024/notes/contents.pdf>, Accessed 1/13, 2011.
- Pipes, L. A., “An Operational Analysis of Traffic Dynamics”. Journal of Applied Physics, Vol. 24, No. 3, pp. 274-281, 1953.
- Rickert, M., Nagel, K., Schreckenberg, M., and Latour, A., “Two Lane Traffic Simulations using Cellular Automata,” Physica A, 231, pp. 534–550, Feb. 1996.
- Salvucci, D. D. and Mandalia, H. M., “Lane-change detection using a computational driver model”. *Human Factors and Ergonomics Society*, 49(3): 532-42, 2007.
- Schakel, W. J., Knoop, V. L. and Van Arem, B., “Integrated lane change model with relaxation and synchronization,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2316, pp. 47–57, 2012.

- Schultz, G. G., and Rilett, L., "Analysis of Distribution and Calibration of Car-Following Sensitivity Parameters in Microscopic Traffic Simulation Models". Transportation Research Record: Journal of the Transportation Research Board, Vol. 1876, No. -1, pp. 41-51, 2004.
- Sparmann, U. "Spurwechselvorgänge auf Zweispurigen BAB Richtungs-fahrbahnen. Forschung Straßenbau und Straßenverkehrstechnik," *Heft*, 1275 vol. 263, 1978.
- Toledo, T., "Integrated driving behavior models," Ph.D. Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Feb. 2002.
- Toledo, T., Koutsopoulos, H. N., and Ben-Akiva, M. E., "Modeling Integrated Lane-changing Behavior," Transportation Research Record, 1857, 30-38, 2003.
- Toledo, T., Koutsopoulos, H. N., and Ben-Akiva, M. E., "Integrated driving behavior modeling," Transportation Research Part C 15 (2), 96–112, Apr. 2007.
- Tomar, R. S., Verma, S., and Tomar, G. S., , "Prediction of Lane Change Trajectories through Neural Network", In IEEE International Conference on Computational Intelligence and Communication Networks, pp. 249-253, 2010.
- Tordeux, A., Lassarre, S., and Roussignol, M., "An Adaptive Time Gap Car-Following Model". Transportation Research Part B: Methodological, Vol. 44, No. 8-9, pp. 1115-1131, 2010.
- Treiber, M., Hennecke, A., and Helbing, D., "Congested traffic States in empirical observations and microscopic simulations," Physical Review E 62.2: 1805, 2000.
- Treiber, M., Hennecke, A., and Helbing, D., "Congested traffic states in empirical observations and microscopic simulations," Physical Review E 62 (2) 1805–1824, Feb. 2000.
- Treiber, M., and Helbing, D., "Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell," 16. Symposium "Simulationstechnik ASIM 2002" Rostock,, edited by Djamshid Tavangarian and Rolf Gr\"utzner pp. 514—520, Sep. 2002.
- Treiber, M., and Kesting, A., "Modeling lane-changing decisions with MOBIL," in: Proceedings of the Traffic and Granularity Flow Conference, Paris, pp. 211–221, June 2007.
- Vahidi, A., and A. Eskandarian. Research Advances in Intelligent Collision Avoidance 36 and Adaptive Cruise Control. Intelligent Transportation Systems, IEEE Transactions on, 37 Vol. 4, No. 3, 2003, pp. 143-153. 38

Wiedemann, R., "Simulation Des straßenverkehrsflusses. Institut für Verkehrswesen der Universität Karlsruhe", 1974.

Yang, Q., and Koutsopoulos, H. N., "A Microscopic Traffic Simulator for Evaluation of Dynamic Traffic Management Systems," *Transportation Research C* 4(3): 113-129, 1996.

## APPENDICES

## Appendix A

### Sample Vehicle Trajectory Dataset

1	483	884	1113433183200.00	17.29100000000000	573.2420000000000	6042777.92800000	2133638.74300000	14.30000000000000	6.40000000000000
2			21.67000000000000	-8.83000000000000	2	3355	11	28.90000000000000	1.33000000000000
1	484	884	1113433183300.00	17.29700000000000	575.3560000000000	6042777.66800000	2133640.84900000	14.30000000000000	6.40000000000000
2			20.88000000000000	-7.31000000000000	2	3355	11	29.64000000000000	1.42000000000000
1	485	884	1113433183400.00	17.30100000000000	577.3870000000000	6042777.41900000	2133642.86200000	14.30000000000000	6.40000000000000
2			20.32000000000000	-4.15000000000000	2	3355	11	30.44000000000000	1.50000000000000
1	486	884	1113433183500.00	17.30500000000000	579.3790000000000	6042777.17700000	2133644.81800000	14.30000000000000	6.40000000000000
2			20.04000000000000	-1.28000000000000	2	3355	11	31.34000000000000	1.56000000000000
1	487	884	1113433183600.00	17.31000000000000	581.3640000000000	6042776.93200000	2133646.80200000	14.30000000000000	6.40000000000000
2			19.97000000000000	0.30000000000000	2	3355	11	32.01000000000000	1.60000000000000
1	488	884	1113433183700.00	17.31500000000000	583.3610000000000	6042776.68700000	2133648.78700000	14.30000000000000	6.40000000000000
2			20	0.37000000000000	2	3355	11	33.31000000000000	1.67000000000000
1	489	884	1113433183800.00	17.32000000000000	585.3650000000000	6042776.44200000	2133650.77200000	14.30000000000000	6.40000000000000
2			20.02000000000000	0.06000000000000	2	3355	11	34.22000000000000	1.71000000000000
1	490	884	1113433183900.00	17.32400000000000	587.3680000000000	6042776.19700000	2133652.75700000	14.30000000000000	6.40000000000000
2			20.02000000000000	0	2	3355	11	34.94000000000000	1.75000000000000

1	491	884	1113433184000.00	17.32800000000000	589.3710000000000	6042775.95200000	2133654.74200000	14.30000000000000	6.40000000000000
	2		20.02000000000000	-0.07000000000000	2	3355	11	35.51000000000000	1.77000000000000
1	492	884	1113433184100.00	17.35800000000000	591.3730000000000	6042775.73200000	2133656.73100000	14.30000000000000	6.40000000000000
	2		20.01000000000000	-0.11000000000000	2	3355	11	36	1.80000000000000
1	493	884	1113433184200.00	17.29000000000000	593.3730000000000	6042775.41500000	2133658.71200000	14.30000000000000	6.40000000000000
	2	20	-0.15000000000000	2	3355	11	36	50000000000000	1.83000000000000
1	494	884	1113433184300.00	17.18900000000000	595.3730000000000	6042775.06600000	2133660.67900000	14.30000000000000	6.40000000000000
	2		19.98000000000000	-0.16000000000000	2	3355	11	37.01000000000000	1.85000000000000

.....continue

878	2660	448	1113433400900.00	55.30300000000000	311.2540000000000	6042848.02000000	2133383.43400000	11.90000000000000	5.30000000000000
	2		37.99000000000000	1.76000000000000	5	860	926	142.40000000000000	3.75000000000000
878	2661	448	1113433401000.00	55.28900000000000	315.0700000000000	6042847.53400000	2133387.23300000	11.90000000000000	5.30000000000000
	2		38.09000000000000	0.16000000000000	5	860	926	142.96000000000000	3.75000000000000
878	2662	448	1113433401100.00	55.29100000000000	318.8920000000000	6042847.06700000	2133391.00400000	11.90000000000000	5.30000000000000
	2		38.07000000000000	-1.13000000000000	5	860	926	143.46000000000000	3.77000000000000
878	2663	448	1113433401200.00	55.29600000000000	322.6880000000000	6042846.60400000	2133394.77000000	11.90000000000000	5.30000000000000
	2	38	-0.21000000000000	5	860	926	143.90000000000000	3.79000000000000	
878	2664	448	1113433401300.00	55.29800000000000	326.4840000000000	6042846.13700000	2133398.54100000	11.90000000000000	5.30000000000000
	2		37.99000000000000	0.11000000000000	5	860	926	144.27000000000000	3.80000000000000

878	2665	448	1113433401400.00	55.30000000000000	330.29000000000000	6042845.670000000	2133402.31200000	11.90000000000000	5.30000000000000
	2		37.94000000000000	-0.890000000000000 5	860	926	144.5800000000000	3.81000000000000	
878	2666	448	1113433401500.00	55.30200000000000	334.0850000000000	6042845.20300000	2133406.08300000	11.90000000000000	5.30000000000000
	2		37.81000000000000	-1.790000000000000 5	860	926	144.8900000000000	3.83000000000000	
878	2667	448	1113433401600.00	55.30400000000000	337.8460000000000	6042844.73600000	2133409.85500000	11.90000000000000	5.30000000000000
	2		37.76000000000000	-0.0900000000000000	5	860	926	145.2500000000000	3.85000000000000
878	2668	448	1113433401700.00	55.30400000000000	341.5950000000000	6042844.26500000	2133413.60800000	11.90000000000000	5.30000000000000
	2		37.98000000000000	4.130000000000000 5	860	926	145.6400000000000	3.83000000000000	
878	2669	448	1113433401800.00	55.31200000000000	345.3830000000000	6042843.79800000	2133417.37800000	11.90000000000000	5.30000000000000
	2		38.58000000000000	8.500000000000000 5	860	926	146.0200000000000	3.78000000000000	
878	2670	448	1113433401900.00	55.35100000000000	349.2750000000000	6042843.35000000	2133421.24300000	11.90000000000000	5.30000000000000
	2		39.45000000000000	9.530000000000000 5	860	926	146.3000000000000	3.71000000000000	



# Sample Customized Vehicle Trajectory Dataset for Linear Model

<i>VID</i>	<i>Relative Velocity</i>	<i>Space Headway</i>	<i>Follower Velocity</i>	<i>Follower Acceleration</i>
1	-2.46678811035157	41.6021709960938	23.3500000000000	0
1	-2.61262453613281	41.3263791748047	23.3500000000000	0
1	-2.62158364257813	41.0605873535156	23.3500000000000	0
1	-2.68153903808594	40.8090037109375	23.4257472167969	1.21195546875000
1	-2.95674350585938	40.5521709960938	23.7067435058594	3.94046103515625
1	-3.32203718261719	40.2385055664063	24.0720371826172	4.22170996093750
1	-3.49000000000000	39.8658810302734	24.2400000000000	0.420163427734375
1	-3.36796281738281	39.4885055664063	24.1179628173828	-3.77091457519531
1	-3.11162824707031	39.1705873535156	23.7900892089844	-4.08021552734375
1	-3.08257993164063	38.9063791748047	23.4858364257813	-1.59352412109375

1	-3.17212639160157	38.5858810302734	23.1932564941406	-3.08866186523438
1	-2.71180666503907	38.2127137451172	22.5249739501953	-8.41413395996094
1	-1.38411147460938	37.9510854980469	21.2961486572266	-11.2000000000000
1	0.507427563476561	37.8994572509766	19.7661486572266	-11.2000000000000
1	2.03873608398438	38.0788698974609	18.5391375244141	-7.80399978027344
1	2.35800742187500	38.4172862548828	18.0742081787109	0.143963037109375
1	1.58343491210938	38.6873308593750	18.1725799316406	3.73682521972656
1	0.424475805664063	38.7631672851563	18.2158364257813	-1.15586640625000
1	-0.507739794921875	38.7021263916016	17.8222602050781	-7.66231979980469
1	-1.04149443359375	38.6121263916016	17.1875092773438	-7.19667641601563
1	-1.43095168457031	38.5095018554688	16.8105427490234	-0.671694970703125

### Sample Generated Reaction time

(Mean 2.2sec and Standard Deviation 0.44)

2.829090889093081

2.157447851638458

1.831983728599337

2.323168017333215

2.328090509450891

3.135942172913503

1.497778576626783

2.018882768722791

2.623841085134539

2.445674401239947

2.282295037010619

2.901763314488963

1.920502595122136

2.406035523044896

2.138595161260279

## Appendix B

### MATLAB Code Generation

#### MATLAB Code for generating random samples using Metropolis-Hasting Algorithm

```
%% Stat meth
clear all;
tic;
str = 'Car_1000_200,000_200_1.txt';

%Input parameter
c1=-0.1052;
c2=-0.0015;
alpha=-0.9424;
beta=1.3930;
gama=-1.2969;

parMean=[-0.1052 -0.0015 -0.9424 1.3930 -1.2969];
ParaVar=[1 1 10 10 10];

Theta=20;

ParamVecAct= [c1 c2 alpha beta gama];
UpdateParamVec=zeros (1, 5);
NewParam=zeros(1,5);
FinalParamVecAct=zeros (100,5);
var=1000000;
burnin=20000;
numiterations=100000;
jumpCovariance = 0.4;

%load DataSet

load Data350.mat;

mem=1;
LM_RMSE_1000(ParamVecAct,Data350);
```

```

actvalue=LM_RMSE_1000(ParamVecAct,Data350);

penatOld=0;
for j=1:5
    penatOld=penatOld+(ParamVecAct(j)-
        parMean(j))^2/(2*ParaVar(j)^2);
end

for i=1:numiterations

    for j=1:5
        NewParam(j)=normrnd(ParamVecAct(j),jumpCovariance);
    end

    newvalue=LM_RMSE_1000(NewParam,Data350);
    penatNew=0;

    for j=1:5
        penatNew=penatNew+(NewParam(j)-
            parMean(j))^2/(2*ParaVar(j)^2);
    end

    Error=(actvalue-newvalue)/2/var;
    Probability=theta*(penatOld-penatNew);

    accept = exp(theta*(penatOld-penatNew)+(actvalue-
        newvalue)/2/var);

    if accept >= 1
        ParamVecAct=NewParam;
        actvalue=newvalue;
        mem=mem+1;
        penatOld=penatNew;
    else
        r=rand(1,1);
        if r<accept
            ParamVecAct=NewParam;
            actvalue=newvalue;
            mem=mem+1;
            penatOld=penatNew;
        end
    end
end

```

```

        end

    end

    if mod(i,100)==0
        i
        mem/100
        jumpCovariance;

        if mem>26.4 && i<200000
            jumpCovariance=jumpCovariance*1.01;
        end
        if mem<20.4 && i<200000
            jumpCovariance=jumpCovariance*.99;
        end
        mem=0;
    end

    FinalParamVecAct(i,:) = ParamVecAct ;
end

X= FinalParamVecAct;
dlmwrite(str,X);
toc;

```

MATLAB Code for calculating sum of square error for 200 observations for 1000 cars

```

function [FinalSumError]=LM_RMSE_1000(ParamVecAct,Data350)

C1= repmat(ParamVecAct(:,1),200,1);
C2= repmat(ParamVecAct(:,2),200,1);
Alpha= repmat(ParamVecAct(:,3),200,1);
Beta= repmat(ParamVecAct(:,4),200,1);
Gama= repmat(ParamVecAct(:,5),200,1);

FinalSumError=0;

for j=1:1000 % Loop for different car

```

```

CarList=(Data350(:,1)==j);
CarSet = Data350(CarList,:);
NewCarSet= CarSet(:,:); % Observation for one car

```

```

% Unknown Values
rel_v= NewCarSet(:,2);
rel_x= NewCarSet(:,3);
vn= NewCarSet(:,4);
accn= NewCarSet(:,5);

```

```

%Sum for each vehicle
accn_hat = dot(C1,rel_v,200)+dot(C2,(rel_x - Alpha -
    dot(Beta,vn,200)- dot(Gama,accn,200)),200);

```

```

% Error for all Car
est_error= dot((accn-accn_hat),(accn- accn_hat),1);

```

```

FinalSumError=FinalSumError+est_error;

```

```

end % for loop j

```

```

end % End of Function

```

#### MATLAB Code for calculating Average mean square error per car

```

%% stat meth
clear all;
tic;
load Data350.mat;

```

```

c1=-0.0969;
c2=0.0172;
alpha=-0.4504;
beta=1.5960;
gama=-1.2387;

```

```

ParamVecAct=[c1 c2 alpha beta gama];

```

```

C1= repmat(ParamVecAct(:,1),200,1);
C2= repmat(ParamVecAct(:,2),200,1);
Alpha= repmat(ParamVecAct(:,3),200,1);
Beta= repmat(ParamVecAct(:,4),200,1);
Gama= repmat(ParamVecAct(:,5),200,1);
e=zeros(0,0);
x=zeros(0,0);
FinalSumError=0;
sumx=0;

for j=1:1000 % Loop for different car

    CarList=(Data350(:,1)==j);
    CarSet = Data350(CarList,:);
    NewCarSet= CarSet(:,j); % Observation for one car

    % Unknown Values
    rel_v= NewCarSet(:,2);
    rel_x= NewCarSet(:,3);
    vn= NewCarSet(:,4);
    accn= NewCarSet(:,5);

    accn_hat = dot(C1,rel_v,200)+dot(C2,(rel_x - Alpha - dot(Beta,vn,200)-
dot(Gama,accn,200)),200);
    PredictedAcc=[NewCarSet(:,5) accn_hat(:,1)];

    error=mean(accn_hat(:,1));
    e=[e;error];

    Acc=mean(NewCarSet(:,5));

    est_error= dot((NewCarSet(:,5)-error),(NewCarSet(:,5)-error),1); %Sum for
each vehicle
    x=[x;est_error];

    FinalSumError=FinalSumError+est_error; % Error for all Car

end % for loop j
x;

```



```
sumx=sum(x(:,1));  
errorpercar=sumx/1000  
toc;
```

### MATLAB Code to calculate model variable the dataset

```
%% meth stat  
clear all;  
tic;  
  
load DataSet.mat;  
load FinalNewDataSet.mat;  
load T.mat  
  
NumberCar=0;  
  
FollowerVelocity200=zeros(0,0);  
SpaceHead200=zeros(0,0);  
FollowerAcc200=zeros(0,0);  
RelativeVelocity200=zeros(0,0);  
  
str = 'FollowerVelocity200.txt';  
str = 'SpaceHead200.txt';  
str = 'FollowerAcc200.txt';  
str = 'RelativeVelocity200.txt';  
  
for j=4:1734 % Loop for diffrent car  
  
    CarList=(FinalNewDataSet(:,1)==j);  
    CarSet = FinalNewDataSet(CarList,:);  
    ZeroLengthCarSet=length(CarSet(:,1));  
  
    if ZeroLengthCarSet==0  
  
        continue  
    else  
  
        NumberCar=NumberCar+1
```

```

for i=1:250 % loop for 400 observation for each vehicle

% Info of Leader Vehicle
ObservationCar=CarSet(i,:);
list=(DataSet(:,1)==ObservationCar(1));
Final_Follower=DataSet(list,:);
ObservationCar(4);
t= ObservationCar(4)-T(j);

% Info of Leader Vehicle
LeaderList=(DataSet(:,1)==ObservationCar(15));
Final_leader=DataSet(LeaderList,:);

Vel_L= interp1(Final_leader(:,4),Final_leader(:,12),t);

% Follower Info
Vel_F=interp1(Final_Follower(:,4),Final_Follower(:,12),t);

Del_x= interp1(Final_Follower(:,4),Final_Follower(:,17),t);
Acc_F=interp1(Final_Follower(:,4),Final_Follower(:,13),t);

% Relative Velocity info

Vel_Rel= Vel_L-Vel_F;

% Unknown variables

FollowerVelocity200=[FollowerVelocity200;Vel_F];
SpaceHead200=[SpaceHead200;Del_x];
FollowerAcc200=[FollowerAcc200;Acc_F];
RelativeVelocity200=[RelativeVelocity200;Vel_Rel];


end % for forloop

end % for if else

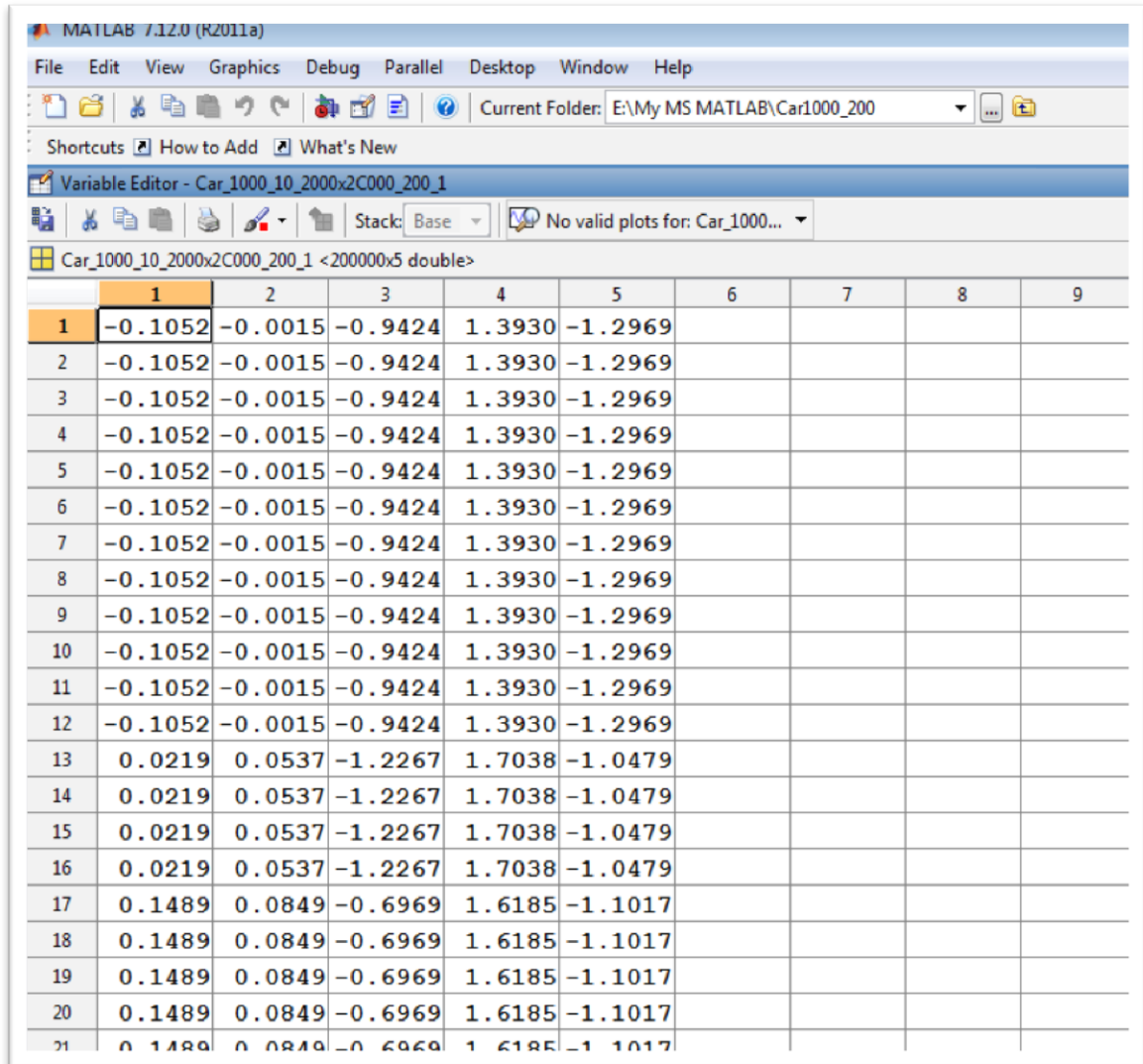
end % for loop j

```

```
dlmwrite(str,FollowerVelocity200);  
dlmwrite(str,SpaceHead200);  
dlmwrite(str,FollowerAcc200);  
dlmwrite(str,RelativeVelocity200);  
  
toc;
```

## Appendix C

### Matlab Output



The image shows the MATLAB 7.12.0 (R2011a) Variable Editor window. The window title is "Variable Editor - Car\_1000\_10\_2000x2C000\_200\_1". The current folder is "E:\My MS MATLAB\Car1000\_200". The variable being edited is "Car\_1000\_10\_2000x2C000\_200\_1", which is a 200000x5 double array. The window displays a table with 20 rows and 9 columns. The first 5 columns contain numerical values, and the remaining 4 columns are empty. The values are as follows:

	1	2	3	4	5	6	7	8	9
1	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
2	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
3	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
4	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
5	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
6	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
7	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
8	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
9	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
10	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
11	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
12	-0.1052	-0.0015	-0.9424	1.3930	-1.2969				
13	0.0219	0.0537	-1.2267	1.7038	-1.0479				
14	0.0219	0.0537	-1.2267	1.7038	-1.0479				
15	0.0219	0.0537	-1.2267	1.7038	-1.0479				
16	0.0219	0.0537	-1.2267	1.7038	-1.0479				
17	0.1489	0.0849	-0.6969	1.6185	-1.1017				
18	0.1489	0.0849	-0.6969	1.6185	-1.1017				
19	0.1489	0.0849	-0.6969	1.6185	-1.1017				
20	0.1489	0.0849	-0.6969	1.6185	-1.1017				
21	0.1489	0.0849	-0.6969	1.6185	-1.1017				